

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

**FERRAMENTA DE ESTRATÉGIA
FINANCEIRA APOIADA POR SISTEMAS
MULTIAGENTES COMPORTAMENTAIS**

Autor: Ramon Cruz da Silva
Orientador: Prof^a.Dr^a. Milene Serrano
Coorientador: Prof.Dr. Maurício Serrano

Brasília, DF
2015



Ramon Cruz da Silva

FERRAMENTA DE ESTRATÉGIA FINANCEIRA APOIADA POR SISTEMAS MULTIAGENTES COMPORTAMENTAIS

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof^a.Dr^a. Milene Serrano

Coorientador: Prof.Dr. Maurício Serrano

Brasília, DF

2015

Ramon Cruz da Silva

FERRAMENTA DE ESTRATÉGIA FINANCEIRA APOIADA POR SISTEMAS MULTIAGENTES COMPORTAMENTAIS / Ramon Cruz da Silva. – Brasília, DF, 2015-

165 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof^ª.Dr^ª. Milene Serrano

Coorientador: Prof.Dr. Maurício Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. Sistemas Multagentes. 2. Bolsa de Valores. I. Prof^ª.Dr^ª. Milene Serrano. II. Prof.Dr. Maurício Serrano. III. Universidade de Brasília. IV. Faculdade UnB Gama. V. FERRAMENTA DE ESTRATÉGIA FINANCEIRA APOIADA POR SISTEMAS MULTIAGENTES COMPORTAMENTAIS

CDU 02:141:005.6

Ramon Cruz da Silva

FERRAMENTA DE ESTRATÉGIA FINANCEIRA APOIADA POR SISTEMAS MULTIAGENTES COMPORTAMENTAIS

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 07 de julho de 2015:

Prof^a.Dr^a. Milene Serrano
Orientador

Prof.Dr. Maurício Serrano
Coorientador

Prof. Dr. Ricardo Matos Chaim
Convidado 1

Prof^a. Dr^a Carla Silva Rocha Aguiar
Convidado 2

Brasília, DF
2015

Dedicatória

Esse trabalho é dedicado aos meus pais, que são os pilares do meu sucesso.

Agradecimentos

Primeiramente agradeço a Deus por ter me proporcionado cursar Engenharia de Software. No início do curso eu não fazia a menor ideia do que é a Engenharia de Software, hoje ela é a paixão que me motiva a buscar o sucesso. Agradeço aos meus pais. Sou grato pela atitude corajosa deles em decidirem sair do interior do estado o Pará e do Maranhão para construir uma família no entorno do Distrito Federal. Se não fosse por essa atitude corajosa, eles não teriam se conhecido e seus filhos não teriam a oportunidade de estudar em uma das 10 melhores universidades do País. Agradeço a eles por me apoiarem e se esforçarem para que eu continuasse estudando nos momentos mais difíceis. Ailton Lima e Maria José, seus esforços não foram em vão, é por vocês que eu vou vencer.

Agradeço ao meu irmão, Renan Cruz, e minha irmã, Nayane Cruz, os quais mostraram-me que irmãos devem ser unidos. Agradeço à minha família e aos amigos que sempre me apoiaram e acreditam em mim. Agradeço à minha namorada Priscila Santa Cruz, que sempre me apoiou e acredita em mim. Agradeço à minha prima Manoela Hungria, que me incentivou ainda na infância a estudar engenharia. Aos meus tios, Maria Cristina e José Ronis, que sempre me apoiaram. Agradeço aos meus amigos da FGA, que compartilharam comigo o mesmo desejo em mergulhar no Mercado Financeiro. Agradeço à professora Milene Serrano e ao professor Maurício Serrano, que aceitaram o desafio em me orientar neste TCC.

*"Combati o bom combate,
completei a corrida,
perseverei minha fé!"
(Bíblia Sagrada, II Timóteo 4, 7)*

Resumo

Segundo uma pesquisa realizada pelo Instituto Rosenfield em 2012, a pedido da BM&FBovespa, apenas 1% da população investe em bolsa de valores. Dentre os motivos elicitados para os que não investem, destacam-se: (i) a falta de conhecimento em como investir, e (ii) a falta de recursos mensais para investir. Diante do exposto, esse Trabalho de Conclusão de Curso (TCC) consistiu no desenvolvimento de uma ferramenta que auxilie o cidadão brasileiro a realizar bons investimentos. Neste TCC, portanto, procurou-se contribuir com os domínios Financeiro e de Engenharia de Software, através da construção de uma ferramenta de estratégia financeira apoiada por Sistemas Multiagentes Comportamentais, onde não é exigido do usuário conhecimentos relacionados ao Mercado Financeiro. Assim, a ferramenta desenvolvida abstrai a complexidade de cálculos financeiros comumente utilizados na análise técnica e delega ao usuário somente a decisão de comprar ou vender. O processo de desenvolvimento da ferramenta foi orientado de acordo com uma adaptação da metodologia Ágil *Scrum* e a implementação foi realizada com a linguagem Java combinada ao framework Jade, adotando ainda boas práticas da Orientação a Objetos visando reduzir o acoplamento e aumentar a coesão do código fonte. Os resultados obtidos foram avaliados de maneira qualitativa através de iterações de pesquisa - ação.

Palavras-chaves: Sistemas Multiagentes, Metodologia de Desenvolvimento Ágil, Mercado Financeiro, Bolsa de Valores.

Abstract

According to a survey provided by Rosenfield Institute in 2012, based on the BM&FBovespa's request, only 1% of the population invests in the stock market. Among the reasons for not investing, there are: (i) lack of knowledge on how to invest, and (ii) lack of resources to monthly invest. As follows, the main goal of this TCC (Course Conclusion Work) is the development of a tool that assists Brazilian citizens on making good investments. The goal of this TCC is to contribute to the financial environment by building a financial strategy tool supported by Multiagent Systems Behavioral, in which is not required Financial Market knowledge for the user. Thus, the tool developed filters financial calculations complexity commonly used in technical analysis and it delegates to the user only the decision to buy or sell. The tool's development was led by an adaptation of the Scrum agile methodology and the implementation was based on the combination of the Java programming language and the JADE framework. Object Orientation good practice was also adopted in order to reduce coupling and increase the cohesion of the source code. Furthermore, the results were qualitatively evaluated through research-action iterations.

Key-words: Multiagent Systems, Agile Software Development, Financial Market, Stock Exchange.

Lista de ilustrações

Figura 1 – Organização do Capítulo	25
Figura 2 – Organização do Capítulo	29
Figura 3 – Composição Mercado Financeiro Brasileiro	30
Figura 4 – Formigueiro	33
Figura 5 – Organização do Capítulo	37
Figura 6 – GrapherOC	38
Figura 7 – MetaTrader	39
Figura 8 – Grafix	40
Figura 9 – <i>Model-View-Controller</i> e o envio de mensagens	41
Figura 10 – Padrão Strategy	42
Figura 11 – Padrão Singleton	43
Figura 12 – Padrão DAO	43
Figura 13 – Arquitetura JADE	44
Figura 14 – Organização do Capítulo	47
Figura 15 – Macro Metodologia	49
Figura 16 – Macro metodologia de desenvolvimento do TCC	51
Figura 17 – Processo investigativo	53
Figura 18 – Candlestick	54
Figura 19 – Padrão DarkCloud	55
Figura 20 – DarkCloud	56
Figura 21 – Scrum adaptado para o TCC	58
Figura 22 – Organização do Capítulo	68
Figura 23 – Camada View	70
Figura 24 – Controller-Model	70
Figura 25 – Pacote core	71
Figura 26 – Estrutura elaborada para execução da Ferramenta em modo offline	71
Figura 27 – Pacote suport	72
Figura 28 – Estrutura desenvolvida para o uso de estratégias financeiras	73
Figura 29 – Processo de montagem de uma carteira	76
Figura 30 – Processo de monitoramento de risco	78
Figura 31 – Média Móvel Simples	79
Figura 32 – Média Móvel Exponencial	80
Figura 33 – Padrão DarkCloud	80
Figura 34 – Padrão Bearish Engulfing	81
Figura 35 – Padrão Bullish Engulfing	82
Figura 36 – Visão geral da ferramenta	84

Figura 37 – Estrutura sugerida para refatoração	85
Figura 38 – Organização do Capítulo	87
Figura 39 – Cobertura de testes unitários	89
Figura 40 – Intervalo para interpretação das métricas.Fonte:(FILHO, 2013)	90
Figura 41 – Questionário aplicado (PARTE I).	92
Figura 42 – Questionário aplicado (PARTE II).	93
Figura 43 – Organização do Capítulo	101

Lista de tabelas

Tabela 1 – Tempo médio de implementação do algoritmo	56
Tabela 2 – Product Backlog inicial	59
Tabela 3 – Cronograma	62
Tabela 4 – Backlog final: histórias de produto concluídas	62
Tabela 5 – Percentual de histórias de pesquisas concluídas	64
Tabela 6 – Percentual geral de conclusão da ferramenta	64
Tabela 7 – Perfis de investidores	73
Tabela 8 – Critérios de aceite de ações	77
Tabela 9 – Estratégias por perfil	82
Tabela 10 – Métricas de código fonte	90
Tabela 11 – Estratégias Perfil Conservador e Resultados	96
Tabela 12 – Estratégias Perfil Moderado e Resultados	97
Tabela 13 – Estratégias Perfil Corajoso e Resultados	97
Tabela 14 – Estratégias Perfil Corajoso e Resultados	98
Tabela 15 – Estratégias Perfil Moderado e Resultados	98
Tabela 16 – Estratégias Perfil Conservador e Resultados	98

Lista de abreviaturas e siglas

Ações	Cotas do capital de uma empresa.
Análise fundamentalista	Estudo realizado com base em informações de mercado e dados contábeis de uma empresa.
Análise técnica	Estudo realizado com base em informações vindas de bolsas de valores (i.e. preço das ações da empresa).
Bacen	Banco Central do Brasil.
BM&Fbovespa	Principal Bolsa de Valores no Brasil.
Constituição Federativa do Brasil	Livro onde são definidos os direitos e deveres de todos cidadãos brasileiros.
CVM	Comissão de Valores Mobiliários.
Deficit	Diferença negativa entre receitas e despesas.
Dividendos	Lucro pago aos portadores de ações de uma determinada empresa.
INSS	Instituto Nacional do Seguro Social.
Superávit	Diferença positiva entre receitas e despesas.
TCC	Trabalho de Conclusão de Curso.
Volatilidade	Variação, instabilidade.

Sumário

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Objetivos	27
1.2.1	Objetivo Geral	27
1.2.2	Objetivos Específicos	27
1.3	Escopo Final do Trabalho	28
1.4	Organização do Trabalho	28
2	REFERENCIAL TEÓRICO	29
2.1	Contexto Financeiro	29
2.1.1	Mercado Financeiro	29
2.1.2	Mercado Capitais	31
2.2	Engenharia de Software	32
2.2.1	Agentes	32
2.2.2	Sistemas Multiagentes	34
2.2.3	Agentes e Objetos	36
2.3	Resumo do Capítulo	36
3	SUPORTE TECNOLÓGICO	37
3.1	Contexto Financeiro	37
3.1.1	Ferramentas de análise técnica	38
3.2	Engenharia de Software	40
3.2.1	Mode-View-Controller (MVC)	40
3.2.2	Padrões de Projeto	41
3.2.2.1	Padrões de projeto adotados neste TCC	41
3.2.3	JADE (<i>Java Agent Development Framework</i>)	44
3.2.4	Metodologia Scrum	45
3.3	Resumo do Capítulo	46
4	METODOLOGIA	47
4.1	Considerações Iniciais	48
4.2	Metodologias de Pesquisa Científica	49
4.3	Classificação da Pesquisa	50
4.4	Metodologia de Condução do Trabalho de Conclusão do Curso	50
4.4.1	Atividade Desenvolver Provas de Conceito	52
4.5	Metodologia Adotada no Desenvolvimento da Ferramenta	58

4.6	Metodologia Adotada na Análise dos Resultados Obtidos	65
4.7	Resumo do Capítulo	65
5	FERRAMENTA FINANCEIRA	67
5.1	Níveis Arquiteturais da Ferramenta	68
5.2	Arquitetura da Ferramenta	69
5.3	Arquitetura da Máquina de Raciocínio dos Agentes	73
5.3.1	Principais agentes	74
5.4	Abordagem para Manutenção Evolutiva da Ferramenta	84
5.5	Resumo do Capítulo	85
6	RESULTADOS	87
6.1	Análise Qualitativa	88
6.1.1	Teste Unitários e Coberturas de Código	88
6.1.2	Métricas de Qualidade de Código	89
6.1.3	Pesquisa - Ação	90
6.1.4	Coleta de Impressões: Usuários	91
6.1.5	Coleta de Impressões: Simulações de Estratégias	95
6.2	Resumo do capítulo	99
7	CONSIDERAÇÕES FINAIS	101
7.1	Principais Contribuições	101
7.2	Limitações	102
7.3	Sugestões de Trabalhos Futuros	102
	Referências	105
	APÊNDICES	109
	APÊNDICE A – APOSENTADORIA NO BRASIL	111
	APÊNDICE B – ALGORITMO DARK CLOUD EM MQL4	115
	APÊNDICE C – ALGORITMO DARK CLOUD EM JAVA + JADE	117
C.1	Classe DarkCloud	117
C.2	Classe abstrata Pattern	118
C.3	Classe CandleStick	118
C.4	Classe LoadCandleStickForCSVFile	120
C.5	Agente experimentalGrafista	121
	APÊNDICE D – ALGORITMO DARK CLOUD EM PYTHON . . .	125

D.1	Bloco DarkCloud	125
D.2	Bloco Bloco FindPatternDarkCloud	126
	APÊNDICE E – RELATÓRIO PESQUISA - AÇÃO 001	127
E.1	Introdução	127
E.2	Participantes	127
E.3	Impressões Coletadas	127
E.4	Ações Tomadas	128
E.5	Conclusão	128
	APÊNDICE F – RELATÓRIO PESQUISA - AÇÃO 002	129
F.1	Introdução	129
F.2	Impressões coletadas	129
F.3	Dados coletados	129
F.3.1	Dados Simulação Perfil Conservador	129
F.3.2	Dados Simulação Perfil Corajoso	146
F.4	Ações tomadas	150
	APÊNDICE G – DOCUMENTO DE ARQUITETURA	153
G.1	Introdução	153
G.2	Objetivos	153
G.3	Arquiterura desenhada	153
G.3.1	Visão Geral	153
G.3.2	Pacote Core	154
G.3.2.1	Agentes implementados	156
G.3.2.1.1	Agente Gestor (Manager)	157
G.3.2.1.2	Agente Especialista (Expert)	159
G.3.2.1.3	Agente Caçador (Hunter)	161
G.3.2.1.4	Agente Criador (Criator)	161
G.3.3	Pacote Suport	162
G.4	Ferramentas	163

1 Introdução

Este capítulo está organizado em seções, como ilustrado na figura 1, e apresenta uma visão geral do contexto onde a ferramenta desenvolvida está situada, bem como uma breve explanação acerca do *framework* JADE, utilizado na investigação do Paradigma de Sistemas Multiagentes. Adicionalmente, são apresentados os objetivos que orientaram o desenvolvimento deste Trabalho de Conclusão de Curso. É apresentado ainda neste capítulo, o escopo final deste trabalho.



Figura 1: Organização do Capítulo

1.1 Motivação

Bolsa de valores, de maneira geral, é um mercado organizado onde uma empresa pública ou privada abre seu capital para arrecadar fundos para fazer investimentos visando crescimento e maior lucro. Ao abrir o capital, a empresa disponibiliza uma quantidade de ações, que podem ser entendidas como pedaços da empresa. Esses, por sua vez, são vendidos para investidores, pessoas físicas/jurídicas ou estrangeiros. Essa empresa assume um compromisso de manter transparência com dados financeiros de interesse dos investidores e pode ser punida por um órgão regulador caso seja identificada alguma irregularidade.

Hoje no Brasil, existe apenas uma bolsa de valores, a Bolsa de Valores de São Paulo, que em 2014, só no mês de fevereiro, movimentou R\$ 131,54 bilhões com uma média de R\$ 6,57 bilhões por dia de pregão. Sendo que dessa movimentação, 50,42% foi realizada por investidores estrangeiros, e apenas 13,73% das movimentações vieram de investidores brasileiros pessoas físicas (BM&FBOVESPA, 2014).

Diante do exposto, percebe-se que o brasileiro não tem uma presença expressiva na bolsa de valores do país. Tal fato é comprovado por um estudo realizado pelo Instituto

Rosenfield, encomendado pela BM&FBovespa referente à educação financeira do brasileiro, o qual revelou que apenas 1% da população investe em ações. Foram ouvidas pelo menos 2.000 pessoas das 15 maiores regiões metropolitanas brasileiras, abrangendo 100 municípios, de todas as classes sociais (ABREU, ABRIL, 2013). O Estudo publicado pela revista RI em 2013 também mostrou que:

1. 40,8% não fazem investimentos. Aplicam em poupança 44,4% dos entrevistados, e 37% deixam o dinheiro na conta corrente. Nem 5% dos entrevistados investem em ao menos um dos demais investimentos, como imóveis, CDB (Certificado de Depósito Bancário), fundos DI (fundos de Depósito Interbancário), ações ou Tesouro Direto.
2. 52,6% dos entrevistados preferem investimentos com baixo risco e baixa rentabilidade, confirmando o perfil conservador do brasileiro, e 65,6% têm pouca propensão para investir em ações.
3. Os dois principais motivos para não investir em ações são a falta de conhecimento (para 43,5% dos entrevistados) e o fato de não disponibilizar de dinheiro ao final do mês (para 21,3% dos entrevistados).
4. Também há bastante desconhecimento sobre a diversificação dos investimentos. Como por exemplo, percebe-se que apenas 22% dos entrevistados sabem que quando se investe em mais de um ativo, o risco de perder dinheiro diminui. (ABREU, ABRIL, 2013)

Diante disto, é possível identificar um *deficit* do cidadão brasileiro em relação aos conhecimentos necessários para investir. Investir na bolsa de valores requer alguns conhecimentos específicos que podem ser automatizados via Software. Um Software capaz de acompanhar e analisar determinados ativos de maneira autônoma, extraindo informações que auxiliem o cidadão a decidir como fazer seus próprios investimentos, de acordo com o seu perfil de investidor. Idealmente, seria como uma ferramenta guia, que mostraria ao cidadão outros tipos de investimentos além das aplicações em poupança. É exatamente essa necessidade que esse Trabalho de Conclusão de Curso procura atender.

Para desenvolver o Software proposto, foi abordado o Paradigma de Sistemas Multiagentes, no intuito de explorar o paradigma, aplicando-o ao contexto financeiro. Para isto foi escolhido o *framework* JADE (*Java Agent Development Framework*).

O JADE é um framework completamente implementado na linguagem Java. Ele simplifica a implementação de sistemas multiagentes através de uma ponte que compila os agentes com as especificações FIPA e oferece suporte de ferramentas gráficas de acompanhamento e possibilidade de realizar atividades de debugging e deploy. A plataforma de agentes é distribuída entre outros computadores (que não necessariamente possuam o mesmo Sistema Operacional) e a configuração pode ser feita através de uma ferramenta remota. Um agente pode ser movido de um computador para outro em tempo de execução sempre que necessário. O JADE é completamente implementado na linguagem Java e requer pelo menos a versão 1.4 do JAVA, JRE ou JDK, instalado. (TELECON, 2014)

Cada agente implementado nesta ferramenta possui um conjunto de comportamentos que define sua especialidade, comportamentos estes que podem ser abstraídos de

investidores humanos existentes no contexto financeiro abordado neste estudo. Adicionalmente, cada agente tem a capacidade de ser autônomo o suficiente para utilizar este conjunto de comportamentos, até mesmo de forma colaborativa com outros agentes, visando alcançar determinados objetivos. No contexto abordado neste trabalho, um objetivo a ser alcançado pelos agentes seria a escolha, por exemplo, dos melhores investimentos alinhados ao perfil investidor do cidadão.

O propósito do Software, desenvolvido como Trabalho de Conclusão de Curso, é oferecer ao cidadão brasileiro uma ferramenta que o auxilie a realizar investimentos em bolsa de valores, de maneira simplificada e apoiada por estratégias financeiras. A ferramenta apresenta opções de investimentos de acordo com o perfil investidor do cidadão. Neste trabalho foram definidos três tipos de perfil investidor: (i) Corajoso, pessoa mais disposta a assumir riscos elevados; (ii) Moderado, pessoa que está entre o perfil Corajoso e Conservador; e (iii) Conservador, pessoa menos disposta a assumir riscos elevados.

1.2 Objetivos

Essa seção compreende os principais objetivos dessa pesquisa, organizados em: objetivo geral (Subseção 1.2.1) e objetivos específicos (Subseção 1.2.2).

1.2.1 Objetivo Geral

Construir uma Ferramenta de estratégia financeira apoiada por Sistemas Multiagentes Comportamentais, trabalhando a máquina de raciocínio dos agentes. Lembrando que a interface propriamente dita não é foco deste Trabalho de Conclusão de Curso, dado que os esforços concentraram-se no raciocínio lógico dos agentes.

1.2.2 Objetivos Específicos

1. Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta.
2. Explorar o Paradigma de Sistemas Multiagentes Comportamentais no contexto financeiro, implementando, por exemplo, comportamentos cíclicos e baseados em tempo.
3. Procurar facilitar a Reutilização de Software, em especial visando melhorar a manutenibilidade da ferramenta bem como sua extensibilidade em trabalhos futuros.
4. Procurar reduzir o acoplamento e aumentar a coesão, apoiando-se em boas práticas da Orientação a Objetos.

1.3 Escopo Final do Trabalho

Ao conduzir o Trabalho de Conclusão de Curso, percebeu-se que o escopo final de trabalho compreende: (i) todo um estudo quanto ao paradigma de Sistemas Multiagentes aplicável ao Mercado de Bolsa de Valores, o qual orientou-se pela modalidade de pesquisa exploratória. Nesse caso, desenvolveu-se uma prova de conceito para embasar as decisões preliminares desse trabalho, e (ii) desenvolvimento propriamente dito da Ferramenta de Estratégia Financeira apoiada por Sistemas Multiagentes Comportamentais bem como a análise qualitativa dessa ferramenta quanto à qualidade do código obtido e as impressões coletadas junto aos potenciais usuários.

Em resumo, o trabalho permitiu estudar uma abordagem emergente, desenvolver um produto de Software, e realizar ciclos evolutivos de teste e refinamento desse produto de Software com base nos resultados obtidos. Resultados esses oriundos da aplicação de pesquisa-ação (ROCHA, 2012) (uma modalidade de pesquisa de natureza qualitativa). Esses aspectos serão principalmente retomados e acordados em detalhes nos capítulos Metodologia, Ferramenta Financeira e Resultados.

1.4 Organização do Trabalho

Este TCC está organizado em Capítulos. No **Capítulo 2**, concentra-se o referencial teórico, este organizado em Contexto Financeiro, com uma breve explanação sobre o mercado abordado, bem como em Engenharia de Software, com uma explanação do Paradigma de Sistemas Multiagentes. No **Capítulo 3**, concentra-se o suporte tecnológico deste TCC, este dividido em Contexto Financeiro, com uma exposição de alguns tipos de ferramentas de Software comumente encontradas no mercado para apoiar atividades de análises técnicas, bem como em Engenharia de Software, com a apresentação dos recursos utilizados neste TCC. No **Capítulo 4**, concentra-se a metodologia adotada. No **Capítulo 5**, concentra-se a descrição da ferramenta desenvolvida. No **Capítulo 6**, são apresentados os resultados obtidos. No **Capítulo 7**, são apresentadas as considerações finais desse estudo.

2 Referencial Teórico

Este capítulo apresenta o suporte teórico utilizado para o desenvolvimento da ferramenta. O capítulo está organizado em seções, conforme ilustrado na figura 2. Na seção 2.1, é apresentado o Contexto Financeiro abordado, bem como a abordagem de análise financeira adotada neste Trabalho de Conclusão de Curso. Na seção 2.2, é apresentada uma descrição do Paradigma de Sistemas Multiagentes explorado. Adicionalmente, é feita ainda neste capítulo uma explanação acerca das três arquiteturas comumente encontradas em Sistemas Multiagentes.

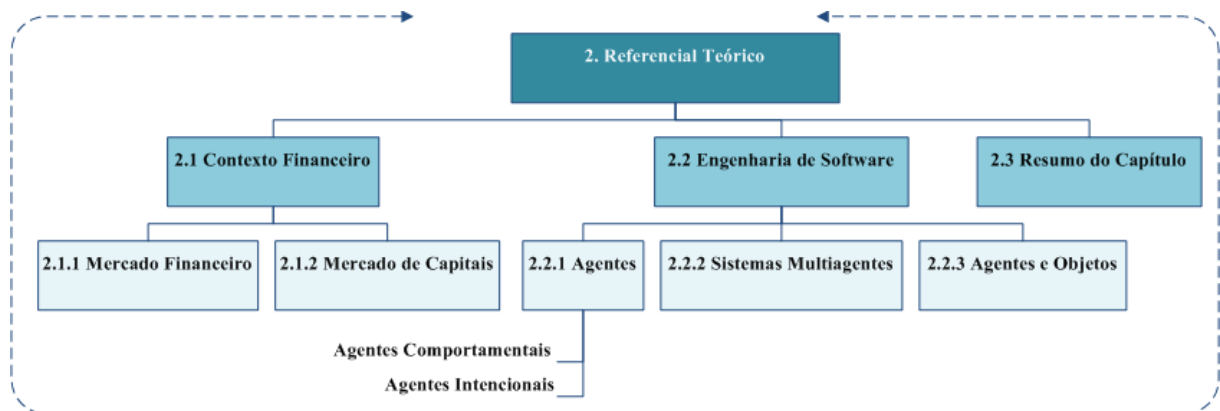


Figura 2: Organização do Capítulo

2.1 Contexto Financeiro

2.1.1 Mercado Financeiro

Em um ambiente capitalista, há o interesse em aplicar recursos financeiros em troca de remuneração. Quando uma pessoa, física ou jurídica, que consome menos do que produz, tornando-se um agente superavitário, deseja investir os valores excedentes, ela procura por outra pessoa, jurídica, que tenha o interesse em captar esses valores em troca de uma remuneração. Essa remuneração é uma taxa de juros que pode ser previamente definida ou não. Assim, quando há pessoas disponibilizando valores e há em contra parte pessoas comprando esses valores, forma-se então um Mercado Financeiro (IMPORTÂNCIA do Mercado Financeiro, 2013).

Segundo a CVM, Comissão de Valores Mobiliários, o Mercado Financeiro Brasileiro é composto por quatro grandes mercados, (CVM, 2014, p. 15), como ilustrado na figura 3 e brevemente apresentados nos parágrafos seguintes.



Figura 3: Composição Mercado Financeiro Brasileiro

Fonte: (CVM, 2014)

O Mercado Monetário é um mercado onde os participantes são instituições financeiras, nas quais há diversas transferências de valores em curto prazo, cerca de um dia. Essas transferências são realizadas entre elas e/ou entre elas e o Banco Central, Bacen. O Banco Central, por sua vez, realiza intervenções neste mercado, visando realizar controles em concordância com a Política Monetária Nacional. Política essa que influencia diretamente no controle inflacionário da economia (CVM, 2014, p. 32).

O Mercado de Crédito é o mercado onde instituições financeiras captam dinheiro de agentes superavitários e realizam empréstimos a pessoas físicas ou jurídicas. Esta instituição tem um compromisso com o agente superavitário comprovado por títulos privados. Ao emprestar um valor à uma pessoa, esta instituição financeira é remunerada com a diferença de juros que cobra do devedor e paga ao seu credor (agente superavitário), denominado *spread*, pelos serviços de intermediação. O Bacen é o principal órgão regulador deste mercado para evitar que ocorram casos de lucros abusivos (CVM, 2014, p. 32).

O Mercado de Câmbio é um mercado onde há transações entre moedas estrangeiras e a moeda nacional. Participam deste mercado instituições que possuem despesas ou receitas em moeda estrangeira. O Bacen é o órgão que regula este mercado visando mantê-lo em concordância com a Política Cambial (CVM, 2014, p. 32).

O Mercado de Capitais é o mercado onde empresas captam valores diretamente com os agentes superavitários, através de títulos como Debêntures, notas promissórias, ações, dentre outros. Em outras palavras, o risco da operação é assumido pelo agente superavitário e não mais pela instituição que faz a intermediação, como no Mercado de Crédito. Porém, essa captação é assessorada por uma empresa apropriada que presta estes

serviços. Este mercado é regulado pela CVM (2014,p 33).Os participantes deste mercado estão apresentados no tópico subsequente, uma vez que este será o mercado abordando neste TCC.

2.1.2 Mercado Capitais

Suponha que uma empresa, cujo nome fantasia seria XPTO S.A, tenha interesse em fazer alguns investimentos em seus projetos. Essa empresa poderá financiar seus investimentos de três maneiras: (i) usando recursos próprios através do lucro acumulado ao longo do tempo; (ii) com base em financiamentos de instituições especializadas em venda de crédito; (iii) ou mesmo através do mercado de capitais, com emissão pública de títulos.

Ao optar pela terceira alternativa, a empresa deverá iniciar, junto com a CVM, um processo de abertura de capital. Depois, deverá atender os critérios estipulados por essa comissão reguladora. Por fim, poderá disponibilizar publicamente suas ações em um Mercado de Capitais.

Ao conseguir a abertura de capital autorizada pela CVM, a empresa emite ações que são adquiridas pelos agentes superavitários, os quais podem ser chamados de investidores, por intermédio de corretoras de valores devidamente registradas. Essas últimas, por sua vez, enviam esses valores à Bolsa de Valores, na qual se encontra a ação disponível. No Brasil, a Empresa responsável por promover um ambiente transparente e líquido para realização dessas negociações é a BM&FBovespa, sediada na capital do Estado de São Paulo.

Para realizar negociações de valores mobiliários, um investidor pode ou não seguir alguma estratégia, que por sua vez pode ou não ser baseada em estudos de especialistas. Estes estudos dividem a atividade de análise de mercado em duas abordagens: a Análise Técnica (NORONHA, 2010) e a Análise Fundamentalista (BUFFET; CLARK, 2010), de um ativo, ambas aplicadas ao Mercado de Ações .

Na abordagem da Análise Técnica, são considerados apenas dados como preço e volume da ação. Em posse destas informações, o analista elabora gráficos, realiza diversos cálculos, usa indicadores matemáticos no intuito de detectar padrões comportamentais do mercado. Vale observar que nesta abordagem não importa a qual empresa pertença a ação, pois o analista deseja encontrar somente os padrões comportamentais dos investidores, para assim elaborar estratégias que retornem o maior lucro possível.

Na abordagem da Análise Fundamentalista, são considerados dados como: (i) Lucro bruto/Margem de lucro bruto; (ii) despesas operacionais; (iii) despesas; (iv) balanço patrimonial entre outros (BUFFET; CLARK, 2010). O analista realiza cálculos para encontrar o valor de mercado da ação desta empresa e o confronta com o valor corrente no mercado, assim realiza operações de compra e/ou venda quando suas estratégias indicam

o melhor momento para operar. Diferente do analista técnico, é importante para o analista fundamentalista conhecer a empresa investida.

São adotadas, neste TCC, técnicas financeiras pertencentes à abordagem da Análise Técnica. Técnicas estas desenvolvidas por especialistas no mercado de capitais e publicadas em mídia impressa e/ou virtual.

2.2 Engenharia de Software

Nesta seção é apresentada uma descrição do Paradigma de Sistemas Multiagentes explorado neste Trabalho de Conclusão de curso, bem como descrição conceitual acerca de Agentes Comportamentais, tipo explorado neste trabalho, e Agentes Intencionais. Adicionalmente, é feita uma explanação acerca das três arquiteturas comumente encontradas em Sistemas Multiagentes (arquitetura quadro-negro, arquitetura de troca de mensagens e arquitetura federativa).

2.2.1 Agentes

Russel e Norving (2003, p. 31) definem: *“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators”*. Em outras palavras, um agente é capaz de se comportar em um determinado ambiente de acordo com as informações oriundas deste ambiente, seja ele determinístico ou não.

Segundo Wooldridge (2002), um agente possui as seguintes características: (i) Autonomia, não há intervenções no processo de decisão de um agente. Ele toma decisões sem intervenções de outros agentes ou humanos; (ii) Sociabilidade, um agente interage com outros agentes, Software ou humano, através de algum tipo de linguagem de comunicação de agentes, e têm a habilidade de trabalhar em conjunto com outros agentes afim de alcançar um objetivo; (iii) Reatividade, um agente fica situado em um ambiente e é capaz de perceber esse ambiente através de seus sensores, sendo capaz ainda de responder em tempo hábil às mudanças que ocorrem nele; e (iv) Pró-atividade, um agente não apenas responde às mudanças do ambiente, ele é capaz de tomar iniciativa (WOOLDRIDGE, 2002, p. 2-3).

De acordo com Girardi (2004, p. 5), o mecanismo utilizado para a tomada de decisão da ação a ser executada a partir de uma percepção é o que distingue um agente, classificando-o em um agente: (i) deliberativo, quando a tomada de decisão da ação a ser executada é realizada via raciocínio lógico em concordância com um plano de ações para atingir alguma meta; (ii) reativo, quando o agente não possui um modelo do ambiente, agindo apenas em resposta a estímulos do ambiente no qual está inserido, sendo assim, um agente reativo não conhece o ambiente ao qual pertence; e (iii) híbrido, quando um agente

possui características de um agente deliberativo e de um agente reativo ([GIRARDI, 2004](#), p. 5).

Agentes Comportamentais: Um agente comportamental é um agente reativo, que tem seus comportamentos abstraídos de comportamentos de uma entidade do mundo real, normalmente ([OLIVEIRA, 2008](#), p. 33). Um conjunto de agentes comportamentais, de maneira cooperativa, pode ser utilizado para solucionar um problema complexo. Um exemplo comumente encontrado na literatura, é o da colônia de formigas. Neste cenário, cada formiga é uma entidade simples, mas em conjunto é capaz de realizar trabalhos complexos existentes em uma colônia, por exemplo: construir um formigueiro semelhante ao da figura 4.



Figura 4: Formigueiro

Fonte: National Geographic

As formigas são divididas por tipos de trabalhos, assim temos as seguintes entidades, ou agentes comportamentais: (i) Rainhas, formigas responsáveis por pôr ovos que darão origem a novas formigas da colônia; (ii) Operárias, formigas que realizam tarefas de acordo com a sua idade e tamanho, e (iii) Machos, formigas que vivem apenas o suficiente para fecundar a Rainha. Em uma colônia, as formigas Operárias mais novas e menores ficam responsáveis por atividades de alimentar a Rainha e lavas, e de limpeza do formigueiro. Já para as operárias mais velhas, ficam as atividades de maior risco, como a proteção do formigueiro (para formigas maiores) e a busca por novas fontes externas de alimento ([HOLDOBLER; WILSON, 1990](#) apud [OLIVEIRA, 2008](#), p. 35).

Alvares e Sichman ([1997](#), p. 12) destacam do exemplo das formigas características dos sistemas multiagentes reativos:

1. Não há representação explícita de conhecimento- o conhecimento dos agentes é implícito e se manifesta através do seu comportamento;
2. Não há representação do ambiente- o seu comportamento se baseia no que é percebido a cada instante do ambiente, mas sem uma representação explícita deste;

3. Não há memória das ações – os agentes reativos não mantêm um histórico de suas ações, de forma que o resultado de uma ação passada não exerce nenhuma influência sobre as suas ações futuras;
4. Organização etológica – a forma de organização dos agentes reativos é similar a dos animais, em oposição à organização social dos sistemas cognitivos;
5. Grande número de membros – os sistemas multiagentes reativos têm, em geral, um grande número de agentes, da ordem de dezenas, centenas ou mesmo milhões de agentes. (ALVARES; SICHMAN, 1997, p. 12).

Agentes Intencionais: Quando um aluno deixa a universidade com uma primeira graduação, ele se depara com uma decisão a tomar, sobre o que fazer com sua vida. O processo de decisão tipicamente inicia-se tentando entender as opções disponíveis em função do conhecimento (crenças) que o aluno possui do seu ambiente. Por exemplo, se o aluno tem um bom histórico escolar, então uma opção é se tornar um acadêmico. Outra opção é se tornar um profissional. Após a geração do grupo de alternativas (desejos), deve-se optar por uma e se comprometer com ela. Esta escolha vem a ser sua intenção. As intenções alimentam o raciocínio lógico futuro do agente. Por exemplo, se o aluno decidir que quer se tornar um acadêmico, então ele deve comprometer-se com esse objetivo e despende tempo e esforço para alcançá-lo. (WOOLDRIGE, 1999 apud GIRARDI, 2004, p. 4).

Agentes deliberativos são agentes capazes de decidir de maneira autônoma seus objetivos e os mudar no decorrer do tempo. Ou seja, estes agentes apresentam uma considerável capacidade cognitiva. Agentes deliberativos possuem as seguintes características:

1. Mantêm uma representação explícita de seu ambiente e dos outros agentes;
2. Podem manter um histórico das interações e ações passadas, isto é, têm memória do passado;
3. A comunicação entre os agentes é feita de modo direto, através do envio e recebimento de mensagens;
4. Seu mecanismo de controle é deliberativo, ou seja, tais agentes raciocinam sobre quais objetivos devem alcançar, quais planos seguir e quais ações devem ser executadas num determinado momento;
5. Seu modelo de organização é baseado em modelos sociológicos, como organizações humanas;
6. Uma sociedade contém tipicamente poucos agentes, na ordem de uma dezena. (FERBER; GASSER, 1991 apud ALVARES; SICHMAN, 1997, p. 28)

Um modelo de agente deliberativo bastante investigado na Inteligência Artificial é o modelo *Belief-Desire-Intention* (BDI) (BRATMAN, 1999 apud SERRANO, 2011, p. 30). Neste modelo, o raciocínio humano é representado através de crenças, desejos e intenções. As crenças formam o conhecimento do agente, e estas são abstrações de informações do mundo real. Os desejos são as metas do agente; são a razão por trás das ações dos agentes. As intenções são as ações do agente; são as sequências de tarefas que o agente desempenha para atender um desejo que está associado às suas crenças (SERRANO, 2011, p. 31).

2.2.2 Sistemas Multiagentes

Girardi (2004, p.6) caracteriza um sistema multiagente (SMA) como *“um grupo de agentes que atuam em conjunto no sentido de resolver problemas que estão além das suas habilidades individuais. Os agentes realizam interações entre eles de modo cooperativo para atingir uma meta.”* Girardi (2004, p.7) define ainda, *“A arquitetura de um SMA mostra a maneira como o sistema está implementado em termos de propriedades e estrutura e como os agentes que o compõe podem interagir a fim de garantir a funcionalidade do sistema.”*

Esta arquitetura pode ser classificada dentre três grupos, conforme sua complexidade: (i) Arquitetura simples, composta apenas por um agente. (ii) Arquitetura moderada, composta por agentes que executam as mesmas tarefas, mas possuem diferentes usuários, sendo que estes agentes podem ainda ser distribuídos em diversos computadores e cooperar entre si. (iii) Arquitetura complexa, composta por agentes de diversos tipos, estes também podem ser distribuídos em diversos computadores e cooperar entre si. As arquiteturas podem ser classificadas considerando seu mecanismo de cooperação e coordenação (KNAPID; JOHNSON, 1998 apud GIRARDI, 2004, p. 7).

Através de um mecanismo de cooperação, os agentes de um SMA expõem suas necessidades para outros agentes. Segundo Girardi, este mecanismo de cooperação é abordado de maneira destacada em três padrões arquiteturais para o desenvolvimento de um SMA: a (i) arquitetura quadro-negro; a (ii) arquitetura de troca de mensagens; e a (iii) arquitetura federativa. (GIRARDI, 2004, p. 7).

Na arquitetura quadro-negro, os agentes se comunicam com outros agentes através de um repositório (organizado em regiões ou níveis para facilitar a busca de informações), que concentra todas as mensagens emitidas por eles, uma espécie de memória compartilhada utilizada por um SMA. Os agentes vão atendendo às mensagens à medida em que vão acessando o quadro-negro (GIRARDI, 2004, p. 7).

Na arquitetura de troca de mensagens, os agentes comunicam-se diretamente uns com os outros através de mensagens assíncronas. Assim, torna-se necessário que cada agente saiba os nomes e endereços dos outros agentes que compõem o SMA. Para que as mensagens ocorram de maneira adequada entre os agentes, é necessário um protocolo de comunicação entre agentes. É no protocolo que estão as regras, as quais agregam um formalismo adequado, e serão seguidas pelos agentes para se obter a melhor comunicação possível (GIRARDI, 2004, p. 8).

Na arquitetura federativa, os agentes são organizados em grupos ou federações. Em cada grupo/federação existe um agente facilitador responsável por receber as mensagens que chega para cada grupo/federação e encaminhá-las para os agentes destinatários presentes no grupo. Ou seja, nesta arquitetura os agentes se comunicam somente através de um agente facilitador (GIRARDI, 2004, p. 8).

2.2.3 Agentes e Objetos

O paradigma de agentes surge como uma evolução natural da orientação a objetos. Como os objetos, a abstração agente possui uma memória e um comportamento. Porém, ela não é uma entidade passiva como os objetos tradicionais. A priori, um agente pode ser definido como um objeto ativo, autônomo, social e com capacidade de aprendizado. (GIRARDI; HYACINTH; WOOLDRIDGE, 2001, 1996, 1999 apud GIRARDI, 2004, p. 2).

De acordo com Wooldridge, Objetos são definidos como entidades computacionais que encapsulam algum estado, são capazes de realizar ações, ou métodos neste estado, e se comunicar por passagem de mensagens. Wooldridge aponta, ainda, três significantes diferenças entre Objetos e Agentes (WOOLDRIDGE, 2002, p. 25-27).

A primeira é o grau de autonomia de Objetos e Agentes, um objeto tem controle sobre seu estado interno, mas não tem controle sobre seu comportamento. Caso um método seja disponibilizado para outros objetos invocarem, eles podem fazer isso a qualquer momento que desejarem. Assim, um objeto não tem controle se o método é executado ou não. Entretanto, um agente ao receber uma requisição, tem o poder de decidir se irá atender ou não. A segunda é a respeito da noção de comportamento autônomo flexível (i.e. reatividade, pró-atividade e habilidade social). A terceira é que cada agente tem sua própria *thread* de controle, (WOOLDRIDGE, 2002, p. 25-27).

2.3 Resumo do Capítulo

Esse capítulo elucidou o Contexto Financeiro, apresentando como o Mercado Financeiro Brasileiro é composto por quatro tipos de mercado (CVM, 2014, p. 15): (i) Mercado Monetário, que é um mercado exclusivo para instituições financeiras; (ii) Mercado de Crédito, mercado onde atuam empresas financeiras especializadas em empréstimos como bancos públicos e privados; (iii) Mercado de Câmbio, onde atuam empresas que normalmente têm despesas e/ou receitas em moeda estrangeira; e (iv) Mercado de Capitais, que é o mercado abordado neste estudo de TCC. Em Engenharia de Software, foi apresentada uma definição de agentes de Software, reforçando suas características quando os mesmos são reativos ou deliberativos. Adicionalmente, ainda sobre Sistemas Multiagentes, comentou-se sobre três arquiteturas comumente encontradas em SMA, tais como: a (i) arquitetura de quadro negro; a (ii) arquitetura de troca de mensagens; e a (iii) arquitetura federativa. O capítulo foi concluído apresentando uma análise comparativa entre as abstrações de agentes e de objeto.

3 Suporte Tecnológico

Este capítulo apresenta o suporte tecnológico utilizado para o desenvolvimento da ferramenta. O capítulo está organizado em seções, conforme ilustrado na figura 5. Na seção 3.1, são apresentadas ferramentas comumente utilizadas em análises técnicas da Bolsa de Valores, vale ressaltar que essas ferramentas compartilham uma característica em comum, que é a necessidade do usuário em possuir conhecimentos avançados acerca de estratégias financeiras. Na seção 3.2, são apresentados os padrões de projetos adotados no desenvolvimento da ferramenta, bem como o padrão arquitetural adotado com propósito de contribuir com a manutenabilidade e extensibilidade da ferramenta. Adicionalmente, ainda nessa seção, é apresentada uma descrição mais detalhada do *framework* JADE. Por fim, é apresentada a metodologia Ágil que foi adaptada para conduzir o desenvolvimento da ferramenta acordada por este Trabalho de Conclusão de Curso.

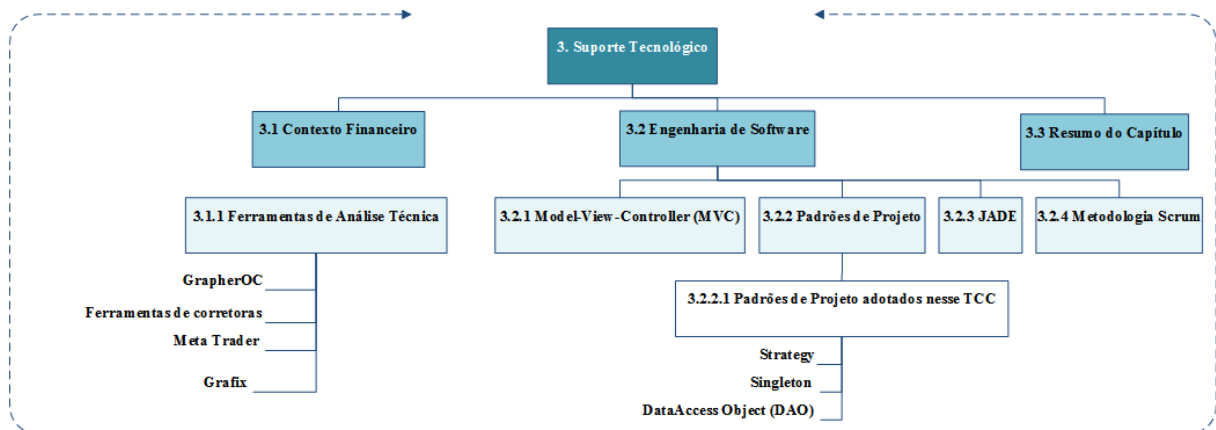


Figura 5: Organização do Capítulo

3.1 Contexto Financeiro

Novos recursos tecnológicos são criados periodicamente, sejam linguagens de programação, novas formas de se solucionar um problema, novos produtos derivados da eletrônica, entre outros. No contexto financeiro não é diferente, ferramentas voltadas ao mercado de capitais, com intuito de automatizar as operações de um investidor neste mercado, encontram-se à disposição.

No entanto, as ferramentas disponíveis ao investidor são ferramentas que o auxiliam na atividade de análise de investimentos, tais como ferramentas de análise técnica. Já outras são serviços oferecidos por *sites* de empresas especializadas, as quais sugerem ao investidor quais papéis são bons de serem comprados ou vendidos. Há ainda ferramen-

tas que apenas auxiliam no acompanhamento da evolução dos investimentos. Percebe-se, então, uma característica em comum entre estas ferramentas, ou seja, é preciso que o investidor tenha bons conhecimentos relacionados ao mercado para que o mesmo possa desfrutar ao máximo dos recursos que essas ferramentas oferecem. Diante disto, a ferramenta desenvolvida nesse TCC abstrai tais conhecimentos e delega ao usuário apenas a decisão em comprar ou vender uma ação. Algumas ferramentas avaliadas neste Trabalho de Conclusão de Curso encontram-se elencadas a seguir.

3.1.1 Ferramentas de análise técnica

GrapherOC: O GrapherOC (figura 6) é um Software de apoio ao acompanhamento do mercado. Com essa ferramenta, o investidor pode testar suas estratégias, obter cotações em tempo real e enviar ordens para corretora. Este Software permite ao investidor criar estratégias e programá-las. No entanto, tal suporte exige que o investidor já possua conhecimentos em estratégias financeiras. Trata-se, por fim, de uma ferramenta disponível somente para computadores com o sistema operacional Windows 7 ®ou superior instalado, não sendo compatível, portanto, com outros sistemas operacionais.



Figura 6: GrapherOC

Ferramentas de corretoras: Corretoras de valores, como a XP Investimentos e Ágora, disponibilizam ao seus clientes ferramentas de envio de ordens de compra e venda (HomeBroker), bem como ferramentas de suporte para atividades de análise técnica. Porém, para usufruir o máximo possível delas, é requerido do usuário conhecimentos financeiros relacionados à análise técnica. No entanto, caso o cliente não tenha tais conhecimentos, as corretoras, normalmente, oferecem acompanhamento de profissionais capacitados e experientes para atuar no mercado.

Meta Trader: O MetaTrader (figura 7) é uma plataforma oferecida por corretoras estrangeiras. Essa ferramenta aborda o mercado de moedas (FOREX) e o de *commodities*, tais como petróleo. A plataforma possui um suporte tecnológico oferecido ao usuário, conferindo ao mesmo uma IDE para desenvolver produtos de Software, conhecidos por robôs, em uma linguagem própria, Mql4 (KOVALYOV, 2000). A linguagem Mql4 é uma linguagem estruturada, baseada na linguagem C, e sua evolução é o Mql5 (METAQUOTES, 2006), a qual é uma linguagem orientada a objetos. No entanto, essa ferramenta requer do usuário conhecimentos relacionados ao Mercado Financeiro, além de conhecimentos de lógica de programação.



Figura 7: MetaTrader

Grafix: O Grafix (figura 8) é uma ferramenta somente para análise gráfica. Tornou-se Software livre no ano de 2012. Trata-se de um suporte construído em Java e seu desenvolvimento está estagnado, desde 2012, por falta de voluntários no desenvolvimento. Assim como as demais, essa ferramenta também exige de seus usuários conhecimentos relacionados a investimentos bem como se diferencia por não ser possível ao investidor enviar ordens de compra/venda diretamente por ela. Limita-se, portanto, apenas em possibilitar a análise do mercado de ações. No entanto, por ser um Software livre, esse suporte permite modificações, sendo possível incorporar novas funcionalidades.



Figura 8: Gráfico

3.2 Engenharia de Software

3.2.1 Mode-View-Controller (MVC)

Segundo Krasner e Pope (1988), quando construímos produtos de Software modularizando seus componentes, obtemos bons resultados. Isolar funcionalmente cada unidade permite aos desenvolvedores entenderem e modificarem mais facilmente cada unidade, sem que seja necessário conhecer todas as unidades envolvidas no Software em questão. Desta forma, uma arquitetura modularizada de um Software contribui para atividades de manutenção e evolução deste, tornando-as mais fáceis (KRASNER; POPE, 1988, p. 1).

A arquitetura *Model-View-Controller* (MVC) concentra-se nos três módulos: (i) *Model*, é a representação o conhecimento que pode ser composto por um ou vários objetos; (ii) *View*, é a representação visual da *Model*, ou seja, é através da *View* que o usuário tem acesso às informações da *Model*; (iii) *Controller*, permite a associação entre *Models* e *Views*, com entradas de dispositivos (i.e. teclados, mouses, entre outros) (KRASNER; POPE, 1988, p. 2).

Para manter todas as *Views* e *Controllers* cientes das mudanças de estado ocorridas na *Model*, foi desenvolvida também a noção de dependência. Nesse contexto, os objetos do tipo *Controller* e *View* são registrados em uma lista de dependentes na *Model*, e quando ocorre alguma mudança, estes são avisados através de troca de mensagens (KRASNER; POPE, 1988, p. 2), como ilustra a figura 9. No capítulo 5, encontra-se uma explanação da aplicação do MVC na ferramenta proposta.

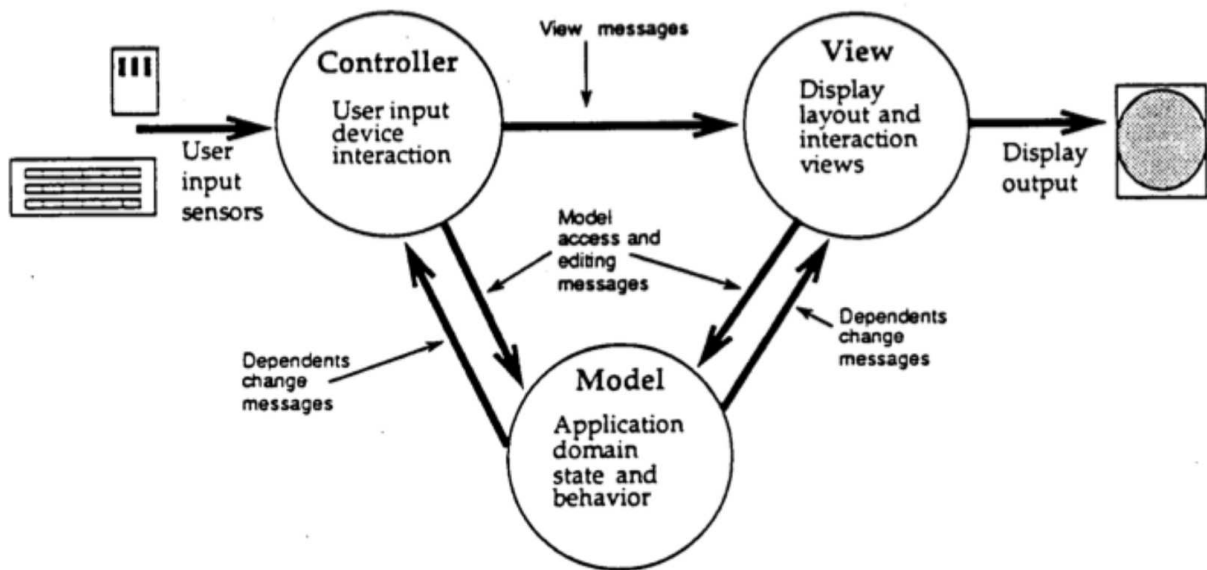


Figura 9: Model-View-Controller e o envio de mensagens

Fonte: (KRASNER; POPE, 1988)

3.2.2 Padrões de Projeto

Christopher Alexander afirma que (ALEXANDER et al., 1977 apud GAMMA et al., 2000, p.19), “cada padrão descreve um problema no nosso ambiente e o cerne da solução, de tal forma que você possa usar essa solução mais de um milhão de vezes, sem nunca fazê-lo da mesma maneira”. Apesar do contexto de Alexander ser o da arquitetura, suas colocações aplicam-se no contexto de desenvolvimento de Software orientado a objetos. O núcleo de ambos os tipos de projetos é a solução para um problema num determinado contexto (GAMMA et al., 2000, p.19).

No contexto de Software, de acordo com Gamma (2000, p.20): “Um padrão de projeto nomeia, abstrai e identifica os aspectos-chave de uma estrutura de projeto comum para torná-la útil para criação de um projeto orientado a objetos reutilizável”. Cada padrão de projeto tem como objetivo solucionar um problema em particular. Para Gamma, um padrão de projeto tem quatro elementos essenciais: (i) Nome do padrão, descreve o problema em uma ou duas palavras; (ii) Problema, descreve a aplicação do projeto; (iii) Solução, descreve os elementos que compõem um projeto, seus relacionamentos, responsabilidades e colaborações; (iv) Consequências, descreve as vantagens e desvantagens de cada padrão (GAMMA et al., 2000, p.19).

3.2.2.1 Padrões de projeto adotados neste TCC

Strategy: Trata-se de um padrão comportamental, o qual permite a variação de algoritmos, independentemente dos clientes que os usam, facilitando ainda ser sensível ao

contexto. O *Strategy* é composto por três participantes: (i) *Context*, é quem chama uma estratégia através de uma entidade do tipo interface; (ii) *Strategy*, é a interface comum a todos os algoritmos relacionados à ela; e (iii) *Concrete Strategy*, é a classe concreta que implementa os diferentes algoritmos (GAMMA et al., 2000, p.294).

Como o padrão *Strategy* delega a implementação de algoritmos às classes concretas bem como se baseia na implementação usando entidades do tipo interface, esse padrão oferece maior facilidade na manutenção de Software, em especial na manutenção evolutiva do mesmo. A figura 10 ilustra a arquitetura básica deste padrão (GAMMA et al., 2000, p.295). Durante o desenvolvimento da ferramenta proposta, outros padrões de projetos foram adotados, conforme a necessidade.

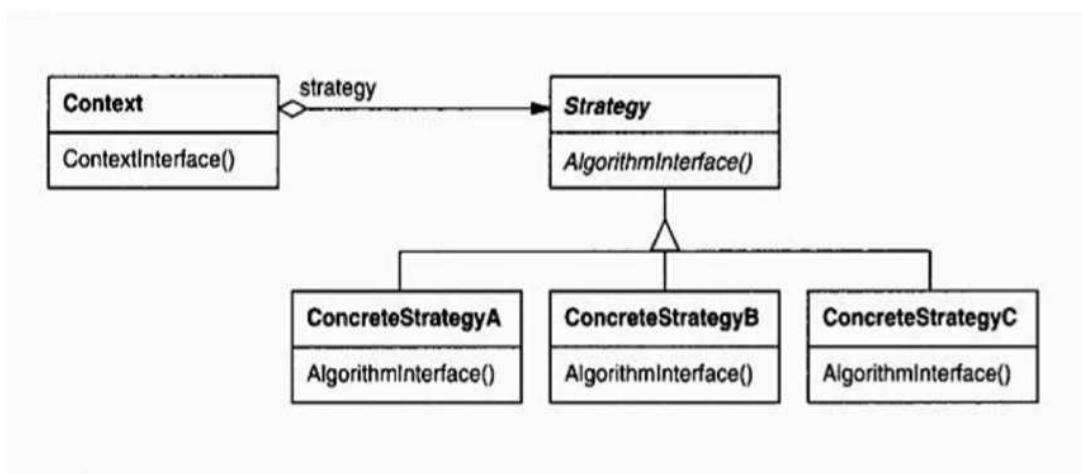


Figura 10: Padrão Strategy

Fonte:(GAMMA et al., 2000)

Singleton: Trata-se de um padrão de criação, cujo objetivo é garantir que uma classe possua somente uma instância e tenha ainda um ponto global de acesso à essa instância. Gamma (2000, p. 130) ilustra um cenário de aplicação deste padrão. Este cenário contém um sistema onde existem várias impressoras e deveria haver somente um *spooler* (i.e. arquivo temporário que concentra os demais arquivos a serem impressos). Da mesma forma, deve haver somente um sistema de arquivos e um gerenciador de janelas. Neste cenário, nota-se a necessidade em garantir que a classe que detem o controle do *spooler* tenha apenas uma instância. E tenha ainda, um ponto público de acesso a essa instância. Gamma (2000, p. 130) afirma que, a melhor solução seria tornar a própria classe responsável por manter o controle da sua única instância. A classe pode garantir que nenhuma outra instância seja criada, bem como pode fornecer um meio para acessar sua única instância. Este é o padrão Singleton.

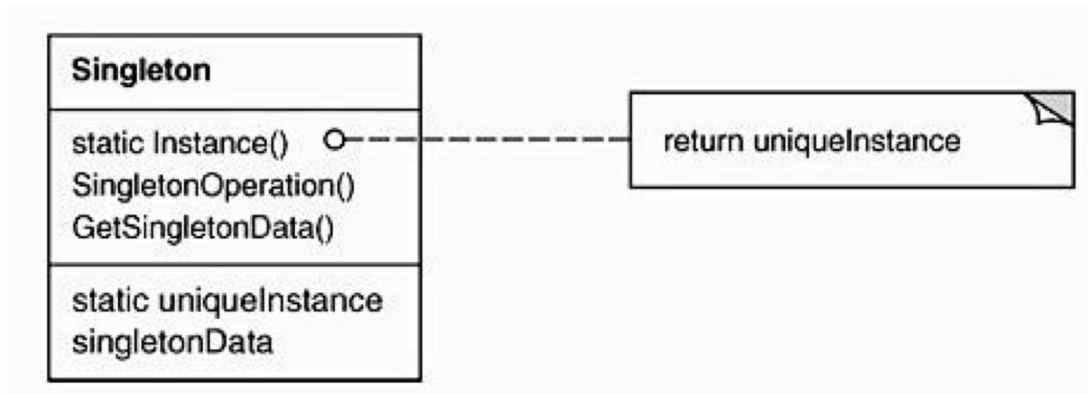


Figura 11: Padrão Singleton

Fonte:([GAMMA et al., 2000](#))

A figura 11 ilustra uma classe que implementa o padrão *Singleton*. Esta classe possui o método estático *Instance()*, que é a única forma de acessar a instância criada por esta classe.

Data Access Object (DAO) Trata-se de um padrão estrutura, cujo propósito é isolar a camada de modelo da camada de persistência. Dessa maneira, obtém-se como benefícios a simplificação da manutenção da camada de persistência, organização na forma de prover e requisitar informações localizadas na base de dados, e baixo acoplamento. A figura 12 apresenta as classes participantes no padrão DAO, tais como: (i) *BusinessObject*, representa as classes da camada modelo; (ii) *DataAccessObject*, representa as classes principais deste padrão. Elas abstraem os acessos aos dados para as classes do tipo *BusinessObject* e permitem o acesso transparente à fonte de dados; (iii) *DataSource*, representa as classes que implementam o acesso aos dados na base de dados do sistema; e (iv) *TransferObject*, representa os objetos utilizados na comunicação entre as classes *DataAccessObject* e *BusinessObject*, ([ORACLE, 2015](#)).

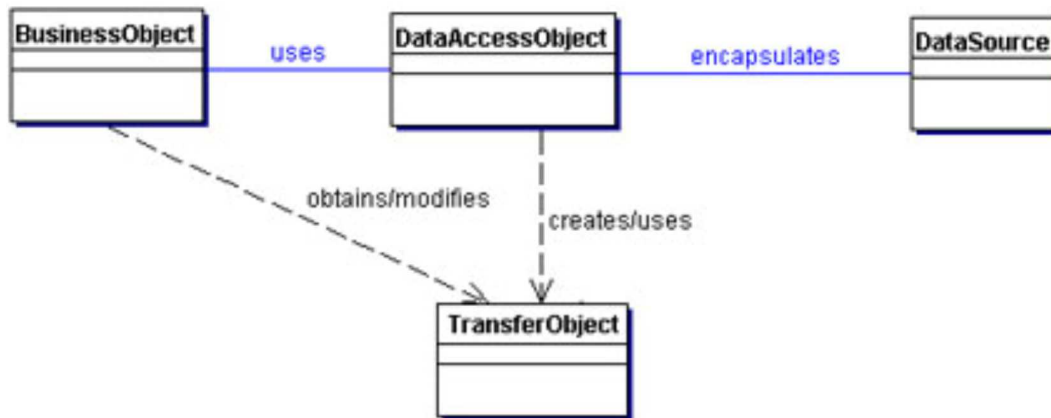


Figura 12: Padrão DAO

Fonte:([ORACLE](#), 2015)

3.2.3 JADE (*Java Agent Development Framework*)

O JADE ([TELECON](#), 2014) é um *framework* implementado em Java que possibilita a programação de Sistemas Multiagentes complexos. O JADE oferece suporte para testes de atividade de *debug*, arquitetura bem estabelecida para criação, comunicação, identificação, gerenciamento e demais ações com vários agentes atuando como em uma sociedade real. No JADE, os agentes podem trocar mensagens entre si, caracterizando sua arquitetura de comunicação como troca de mensagem. Este *framework* foi desenvolvido de acordo com as especificações da *Foundation for Intelligent Physical Agents* (FIPA) ([TELECON](#), 2014), que é uma comunidade IEEE que promove padrões tecnológicos para Sistemas Multiagentes. Por ter sido implementado integralmente em Java, o JADE pode ser executado em qualquer sistema operacional que possua uma Máquina Virtual Java.

O JADE é um Software livre e distribuído pela Telecom Itália, empresa proprietária da TIM Brasil, e possui a Licença LGPL ([TELECON](#), 2014). Ele é supervisionado por um grupo de empresas de cinco membros: Telecom Itália, Motorola, Whitestein Technologies AG, ProfactorGmbH e France Telecom R&D. Atualmente, a versão disponível é a 4.3.3, publicada em 11 de dezembro de 2014.

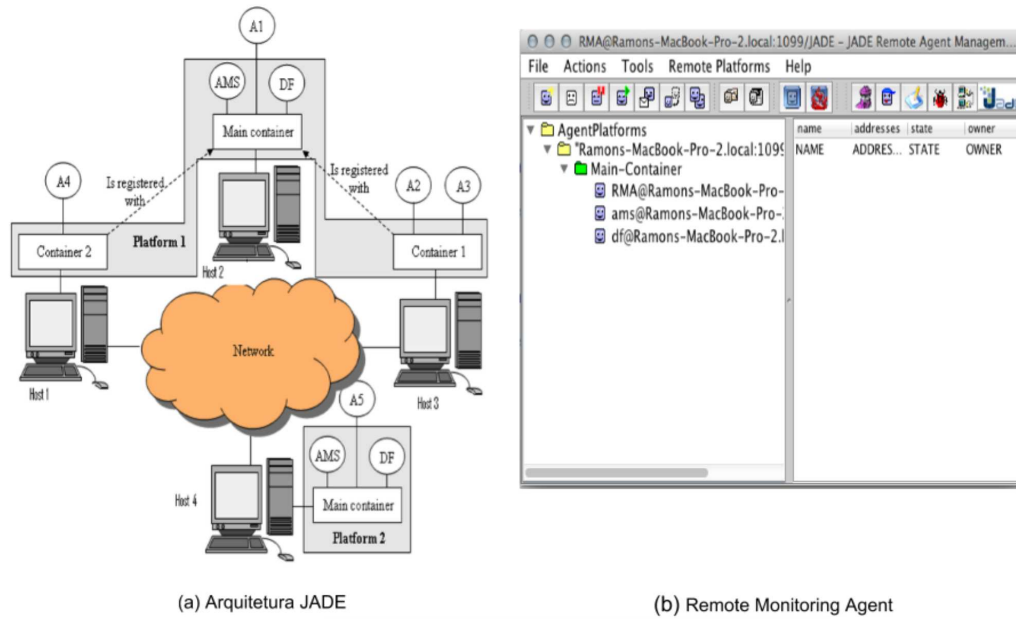


Figura 13: Arquitetura JADE

Fonte: (TELECON, 2014)

Considerando a Arquitetura da Plataforma JADE apresentada na figura 13.a, cada Agente instanciado fica contido em um *Container*, sendo que este pode ter um ou vários agentes. Existem dois tipos de *containers*, o main e o padrão. O *main container* contém, por padrão, três agentes: (i) Agente DF (*Directory Facilitator*), agente responsável por prover serviços, conhecidos no JADE como páginas amarelas, nos quais um dado agente pode encontrar outros agentes, iniciar comunicação com terceiros e registrar/cadastrar os serviços que presta a terceiros; (ii) Agente AMS (*Agent Management System*), agente responsável por prover serviços de páginas brancas, sendo capaz de criar e excluir agentes em containers remotos ou locais, gerenciando o ciclo de vida dos agentes na plataforma como um todo; e (iii) Agente RMA (*Remote Monitoring Agent*) (figura 13.b), agente responsável por gerenciar as consoles gráficas providas pela Plataforma JADE.

Simplificadamente, na Arquitetura do JADE, uma dada plataforma é composta por um ou vários *containers*. Estes *containers* podem estar em um computador ou distribuídos em uma rede de computadores, como ilustrado na figura 13a. A arquitetura JADE possibilita ainda a comunicação entre os agentes independentemente do *container* onde estes se encontram, ou seja, o agente A2 pode se comunicar com o agente A3, assim como pode se comunicar também com o agente A5, o qual está em outro *container*, em outra plataforma, porém conectado a uma rede de computadores. Adicionalmente, serviços, conteúdos e servidores podem ser compartilhados usando como base essa rede de computadores, permitindo aos agentes trabalharem de forma colaborativa, conquistando objetivos até mesmo não triviais, centrados no princípio Dividindo para Conquistar. Princípio esse oriundo de contextos político-sociais bem como amplamente utilizado na atualidade, na

ciência da computação como um todo, como uma técnica capaz de lidar com problemas complexos, dividindo esse problema em instâncias menores, resolvendo essas instâncias isoladamente e agrupando os resultados para atingir uma conquista maior, ou seja, a solução do problema de maior complexidade.

3.2.4 Metodologia Scrum

O termo *Scrum* é originário do *Rugby*, esporte criado na Inglaterra. No *Scrum*, são feitas entregas contínuas de pequenas partes funcionais do Software. O Scrum consiste em times associados a papéis, eventos, artefatos e regras. Cada componente tem uma função específica no desenvolvimento do Software (SCHWABER; SUTHERLAND, 2013).

O time *Scrum* é formado por: (i) um *Scrum Master*, que é a figura do líder do time. Ele é o responsável por fazer o time compreender as regras do *Scrum*; (ii) um grupo de desenvolvimento (time), que são os programadores, designers, entre outros; e (iii) um *Product Owner*, que é uma pessoa com bom conhecimento das necessidades do produto. No *Scrum*, os times são auto-organizáveis no que tange à escolha da melhor forma para realizar um trabalho. O modelo de time no *Scrum* foi pensado de forma que valorize a flexibilidade, criatividade e produtividade. O time entrega produtos de forma iterativa e incremental, assim aumentando oportunidades de *feedback* contínuo. Estas entregas são versões parciais funcionais do Software trabalhado pelo time (SCHWABER; SUTHERLAND, 2013, p. 5-7).

Cada evento no *Scrum* é uma oportunidade de verificar e corrigir algo no produto. Eles são projetados para que seja possível obter transparência e uma inspeção criteriosa do produto. Cada evento termina sempre que seu objetivo é alcançado. No *Scrum*, o evento principal é a *Sprint*. Ela é o ciclo de desenvolvimento do produto, que dura um mês ou menos, sendo composta por: (i) uma reunião de planejamento; (ii) reuniões diárias curtas; (iii) uma revisão da *Sprint*, (iv) uma retrospectiva; e o (v) trabalho de desenvolvimento do produto.

Segundo Schwaber e Sutherland (2013, p.13), “Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidade para inspeção e adaptação.” Os artefatos são elaborados de forma que o entendimento das informações contidas nele seja o maior possível. Dentre os artefatos do Scrum temos: (i) *Backlog* do Produto, é uma lista de tudo que é necessário no produto. Ela é a origem dos requisitos e é gerida pelo *Product Owner*. O *Backlog* do Produto é dinâmico e existe enquanto houver o produto; (ii) *Backlog* da *Sprint*, é o conjunto de itens selecionados do *Backlog* do Produto para a *Sprint*, ou seja, é o conjunto de características que serão implementadas pelo time durante a *Sprint*; (iii) Incremento, é a soma de todos os itens do *Backlog* do Produto entregues no final de uma *Sprint* (SCHWABER; SUTHERLAND, 2013, p. 13-15).

3.3 Resumo do Capítulo

Elucidou-se nesse capítulo, no Contexto Financeiro, uma breve exposição de algumas ferramentas de análise técnica comumente encontradas no Mercado Financeiro para apoiar o investidor. Destaca-se nelas uma característica em comum, que é a necessidade de seu usuário possuir conhecimentos relacionados à análise técnica.

Na Engenharia de Software, foi apresentado um modelo arquitetural bastante aceito, o *Model-View-Controller*. Este modelo concentra-se na divisão da arquitetura de um Software em três módulos autônomos que se comunicam entre si. Foram apresentados ainda os padrões de projeto: (i) *Strategy*, (ii) *Singleton*, e (iii) DAO. Adotados neste TCC.

Adicionalmente, foi descrito o *framework* JADE, o qual foi adotado neste TCC para implementar os agentes comportamentais planejados para o desenvolvimento da ferramenta. E por fim, foi apresentada a metodologia Ágil, que foi adaptada para conduzir o desenvolvimento da ferramenta acordada por este TCC.

4 Metodologia

Neste capítulo, serão apresentadas as principais metodologias de pesquisa científica encontradas na literatura. Em um segundo momento, será especificada qual a metodologia de pesquisa científica, dentre as pesquisadas, que mais se adequou ao tema coberto nesse Trabalho de Conclusão de Curso. Posteriormente, são descritas: (i) a metodologia utilizada para condução do Trabalho de Conclusão de Curso como um todo, da atividade de levantamento de referencial teórico à escrita final do trabalho, chamada macro metodologia. Ainda nessa seção, será descrita a atividade de desenvolvimento de provas de conceito, a qual foi de suma importância para definir o suporte tecnológico adequado ao desenvolvimento da ferramenta; (ii) a metodologia que conduziu o desenvolvimento da ferramenta em si, e (iii) a metodologia que orientou a análise dos resultados obtidos. A organização deste capítulo é ilustrada na figura 14.

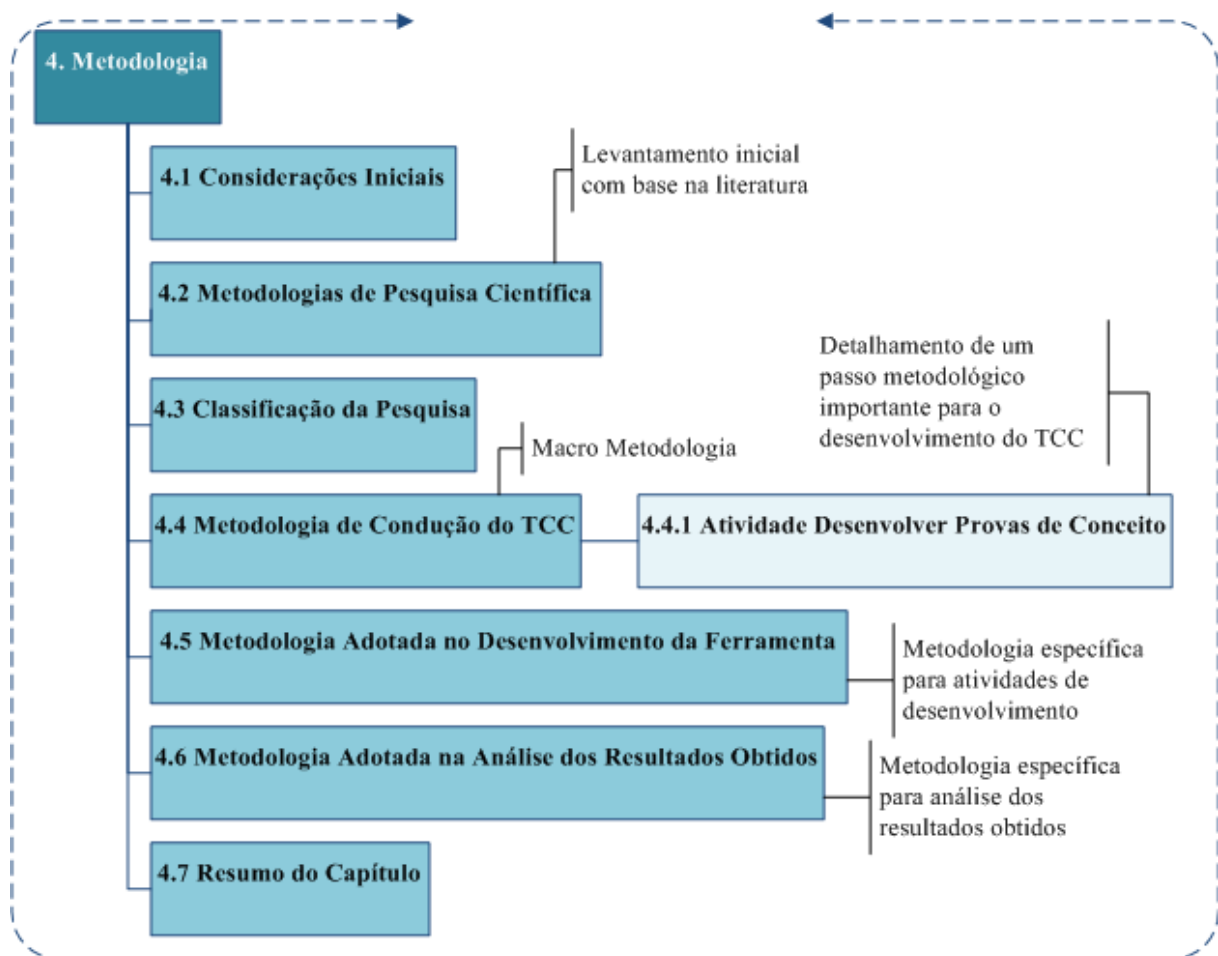


Figura 14: Organização do Capítulo

4.1 Considerações Iniciais

Esse trabalho compreendeu principalmente as seguintes atuações:

- explorar o paradigma de Sistemas Multiagentes, estabelecendo a viabilidade de se utilizar esse paradigma no Mercado de Bolsa de Valores usando como base uma prova de conceito (essa descrita na seção 4.3.1 desse capítulo). Nesse caso, foi utilizada predominantemente a modalidade de pesquisa científica exploratória, cujas razões para essa classificação serão apresentadas nas seções 4.1 e 4.2 desse capítulo. Pode-se adiantar que o paradigma em questão é intrinsecamente emergente, justificando naturalmente o uso de uma abordagem exploratória;
- desenvolver a ferramenta em si, ou seja, uma Ferramenta de Estratégia Financeira apoiada por Sistemas Multiagentes Comportamentais. Nesse caso, por se tratar de um produto de Software, foi utilizada uma metodologia específica para condução desse desenvolvimento, a qual será apresentada na seção 4.4 desse capítulo, e
- analisar qualitativamente o produto de Software obtido, ou seja, a ferramenta desenvolvida. Nesse caso, conduziu-se essa análise de resultados com base em uma metodologia de pesquisa específica para esse fim. Essa será apresentada na seção 4.5 desse capítulo.

Vale ressaltar ainda que o Trabalho de Conclusão de Curso, por envolver outras atividades compreendidas entre o levantamento de referencial teórico e a escrita final do TCC, orientou-se por uma macro metodologia, a qual será descrita na seção 4.3 desse capítulo. Essa metodologia contempla uma sequência de atividades. Dentre essas atividades é possível identificar, por exemplo, as atividades dedicadas à realização da prova de conceito; ao desenvolvimento da ferramenta em si e à análise dos resultados obtidos com a coleta de impressões da ferramenta, conforme ilustrado na figura 15.

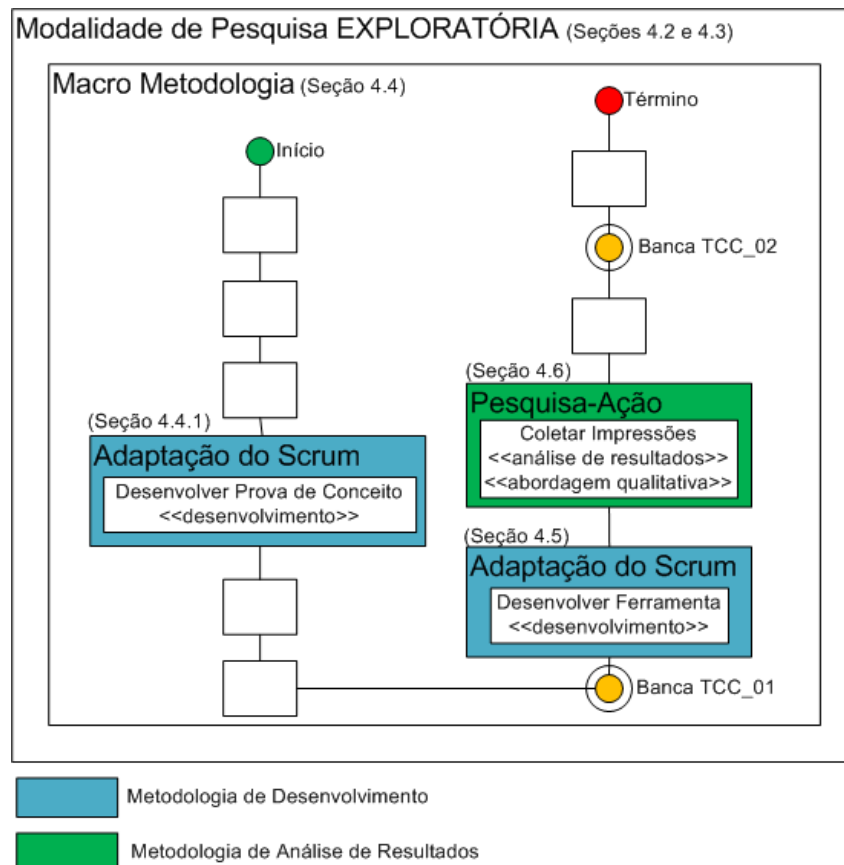


Figura 15: Macro Metodologia

4.2 Metodologias de Pesquisa Científica

Segundo Gil (2002, p.17), pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. Gil afirma ainda, que é usual classificar uma pesquisa com base em seus objetivos gerais. Assim, uma pesquisa pode ser classificada em três grupos: (i) exploratórias; (ii) descritivas; e (iii) explicativas. (GIL, 2002, p. 41).

As pesquisas exploratórias têm como objetivo principal buscar familiaridade com o problema abordado, de maneira a aprimorar ideias. Pesquisas exploratórias possuem um planejamento mais flexível. Entretanto, na maioria dos casos assume a forma de pesquisa bibliográfica ou de estudo de caso. (GIL, 2002, p. 41).

As pesquisas descritivas têm como objetivo principal descrever características de determinada população, fenômeno, ou a descrição de relações entre variáveis. Segundo Gil (2002, p.42), as pesquisas descritivas e as exploratórias são habitualmente realizadas por pesquisadores sociais preocupados com a atuação prática e, geralmente, assumem a forma de levantamento.

As pesquisas explicativas têm como objetivo principal a identificação dos fatores que determinam ou contribuem para ocorrência de um determinado fenômeno. De acordo

com Gil (2002, p.42), esse é o tipo de pesquisa mais complexo e delicado, pois é exigido um profundo conhecimento da realidade. No entanto, isso não significa que as pesquisas exploratórias e descritivas tenham um valor inferior. Estas, geralmente, constituem uma etapa prévia indispensável para que se possa obter explicações científicas.

O levantamento acordado anteriormente procura classificar a pesquisa científica quanto aos seus objetivos. Com base no tipo de pesquisa científica, algumas metodologias podem ser mais adequadas do que outras. Por exemplo, caso a pesquisa seja de cunho exploratório, cabe a utilização de uma metodologia de pesquisa científica exploratória.

Adicionalmente, uma pesquisa pode ser classificada quanto à sua abordagem. Nesse caso, uma pesquisa pode ser quantitativa (aplicada quanto se pode mensurar os dados analisados), qualitativa (quando os dados analisados são subjetivos) ou híbrida. Nesse último caso, a abordagem envolve ambas as análises, quantitativa e qualitativa.

4.3 Classificação da Pesquisa

O tipo de pesquisa adotado neste TCC foi o exploratório, cujo objetivo foi explorar o paradigma de Sistemas Multiagentes Comportamentais no contexto Financeiro, especificamente no contexto da Bolsa de Valores de São Paulo. O resultado desta pesquisa foi uma ferramenta de estratégia financeira capaz de abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. A ferramenta foi projetada de maneira a facilitar a evolução da máquina de raciocínio dos agentes, bem como suas futuras manutenções.

4.4 Metodologia de Condução do Trabalho de Conclusão do Curso

A realização do TCC foi orientada por uma macro metodologia, a qual foi organizada em atividades. Essas atividades compreenderam desde "Levantar Referencial Teórico" a "Escrever TCC 02", conforme apresentado na figura 16, modelada no Bizagi ©.

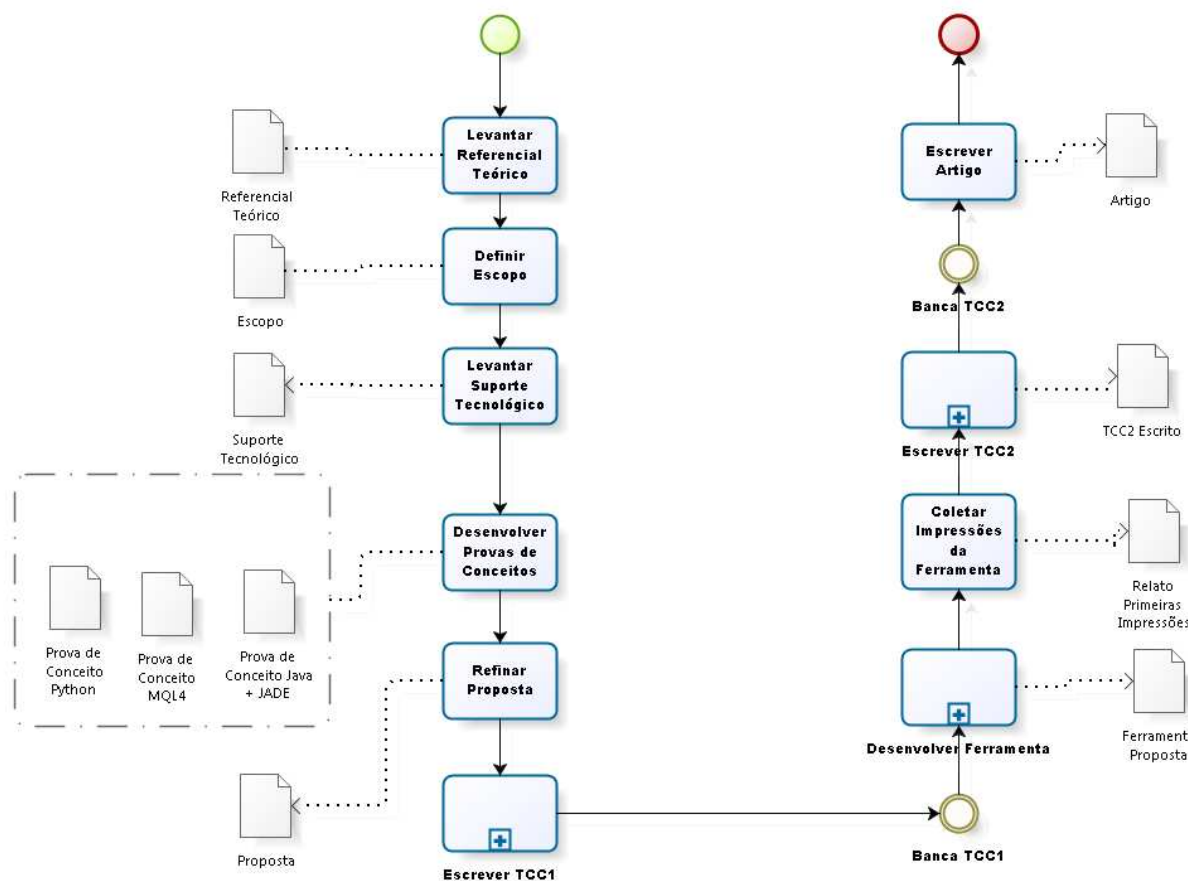


Figura 16: Macro metodologia de desenvolvimento do TCC

Visando elucidar o que foi realizado em cada atividade, seguem breves descrições sobre essas atividades:

Levantar Referencial Teórico: Nesta atividade, foram feitos levantamentos de artigos científicos, artigos jornalísticos, publicações de empresas públicas/privadas, possíveis tecnologias que mais se adequassem ao tema tratado no trabalho, possibilitando maior conhecimento dos domínios Financeiro e de Engenharia de Software.

Definir Escopo : Nesta atividade, o escopo do TCC foi definido como sendo "Construir uma Ferramenta de Estratégia Financeira apoiada por Sistemas Multiagentes Comportamentais".

Levantar Suporte Tecnológico: Nesta atividade, foram levantadas as tecnologias que mais se adequavam, em um primeiro momento, ao tema coberto no TCC.

Desenvolver Provas de Conceito: Nesta atividade, foram implementadas provas de conceito, utilizando uma adaptação da metodologia *Scrum*, em diferentes suportes tecnológicos, visando à definição de uma boa estratégia em atendimento à ferramenta.

Refinar Proposta: Nesta atividade, foi feita uma revisão da proposta quanto ao Referencial Teórico, ao Suporte Tecnológico bem como outros detalhes, em atendimento ao tema do trabalho.

Escrever TCC1: Nesta etapa, foi feita a documentação dos resultados obtidos nas etapas anteriores, possibilitando a obtenção da escrita final do TCC01.

Após a avaliação da Banca de TCC1, ajustes foram realizados na parte escrita, no intuito de atender as colocações apontadas pelos membros da Banca. Posteriormente, o TCC foi conduzido considerando as seguintes atividades metodológicas:

Desenvolver Ferramenta: Nesta atividade, foi feito o desenvolvimento da ferramenta utilizando uma adaptação da metodologia Scrum. Os passos metodológicos dessa atividade são descritos na subseção 4.5, ainda neste capítulo.

Coletar Impressões da Ferramenta: Um vez desenvolvida a primeira versão da ferramenta, foram conduzidos testes em laboratório bem como junto aos usuários finais da ferramenta. Esses testes possibilitaram análises quantitativas e qualitativas, permitindo ainda a coleta das primeiras impressões da ferramenta. Com base nesses resultados, refinamentos e ajustes foram realizados usando uma abordagem de pesquisa-ação. Parte desses refinamentos foram acordados de forma empírica. Esse processo foi iterativo-evolutivo, sendo realizadas, portanto, algumas iterações até a obtenção da versão atual da ferramenta. Todo esse processo é apresentado em detalhes no capítulo 6 desse documento.

Escrever TCC2: Nesta atividade, foram feitas as subatividades: (i) Revisar e refinar Referencial Teórico; (ii) Documentar o desenvolvimento da ferramenta; (iii) Coletar e analisar as impressões da ferramenta; (iv) Documentar o TCC2, em conformidade com as normas ABNT.

Escrever Artigo: Nesta atividade, será elaborada a escrita de um artigo para publicação em eventos relacionados ao desenvolvimento de sistemas multiagentes.

Uma atividade que se destaca dentre as apresentadas foi a atividade intitulada "Desenvolver Provas de Conceito". Na próxima subseção, serão apresentados detalhes sobre esse desenvolvimento, o qual foi de suma importância para uma visão preliminar da ferramenta e do suporte tecnológico utilizado ao longo do TCC.

4.4.1 Atividade Desenvolver Provas de Conceito

Nesta etapa do TCC, foi elaborado um cenário para avaliar o esforço necessário do autor para implementar um Software nas linguagens Java, Mql4 e Python. Para isso,

foram adotadas as seguintes etapas para realizar uma prova de conceito: (i) Elaborar cenário, onde foi eleito um padrão de formação encontrado nos gráficos de *candlesticks* e foi elaborada uma hipótese inicial; (ii) Selecionar ativos, onde foram eleitas algumas ações com seus devidos valores históricos para testar os produtos de Software; (iv) Implementar, onde os algoritmos foram implementados nas três linguagens escolhidas; (v) Analisar resultados, onde foi feita a análise manual dos resultados afim de verificar o grau de acerto. Estas etapas estão descritas nos subtópicos seguintes e ilustradas na figura 17.

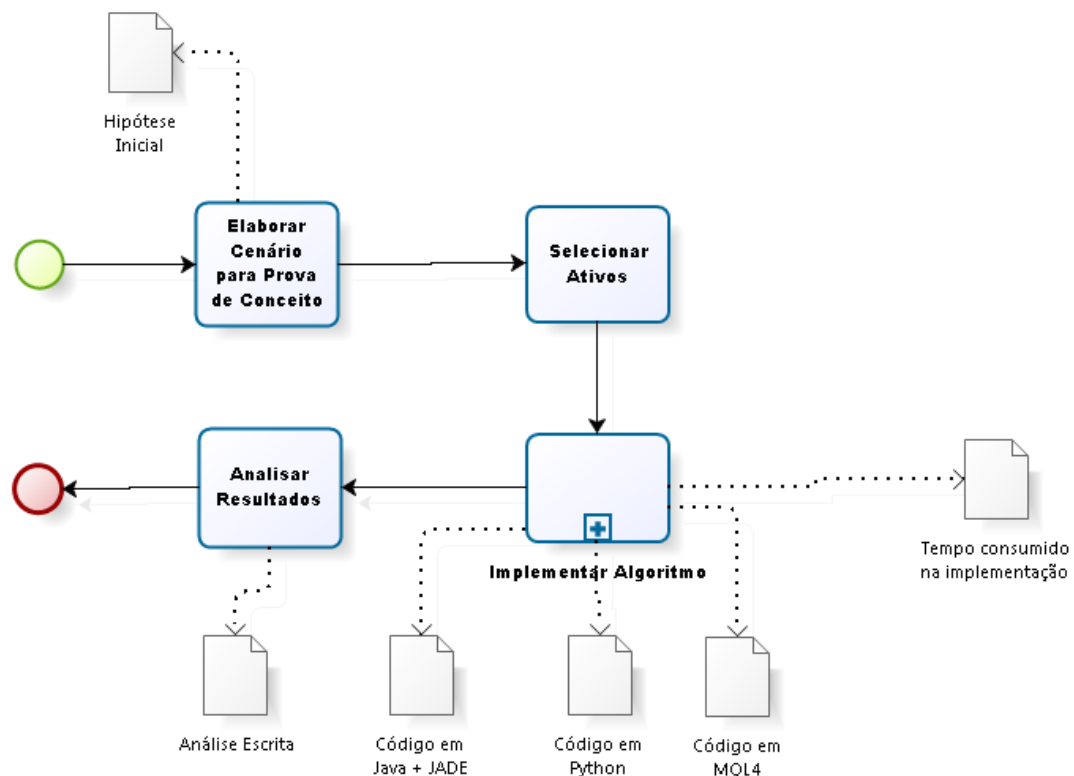


Figura 17: Processo investigativo

Elaborar Cenário para Prova de Conceito: Para nortear as provas de conceito feitas para este estudo, foi elaborado um cenário onde é requerida uma maneira automatizada de se chegar à uma solução. Este cenário é descrito a seguir.

Uma das maneiras de um investidor acompanhar as movimentações do mercado de capitais, mais especificamente uma Bolsa de Valores, é através de gráficos como: (i) o gráfico de barras ; (ii) o gráfico de linha e (iii) o gráfico de candlestick(MATSURA, 2006, p. 5-6). Considerando o último, deseja-se encontrar uma maneira automatizada bem como a formação do padrão de candlestick conhecido como Dark cloud (MATSURA, 2006, p.61). Padrão este, que poderá compor um plano estratégico de investimento.

Uma *candlestick* é formada por 4 valores: (i) valor de abertura; (ii) valor de máxima; (iii) valor de mínima; e (iv) valor de fechamento. Quando o valor de fechamento é maior do que o valor de abertura tem-se uma *candlestick* de alta, quando ocorre o inverso tem-se uma *candlestick* de baixa. A figura 14 ilustra uma *candlestick* de alta, que é representada com o preenchimento em branco e uma *candlestick* de baixa, que é representada com o preenchimento em vermelho. Cada *candlestick* representa uma periodicidade, a qual pode ser desde um minuto até um ano (MATSURA, 2006, p.6).

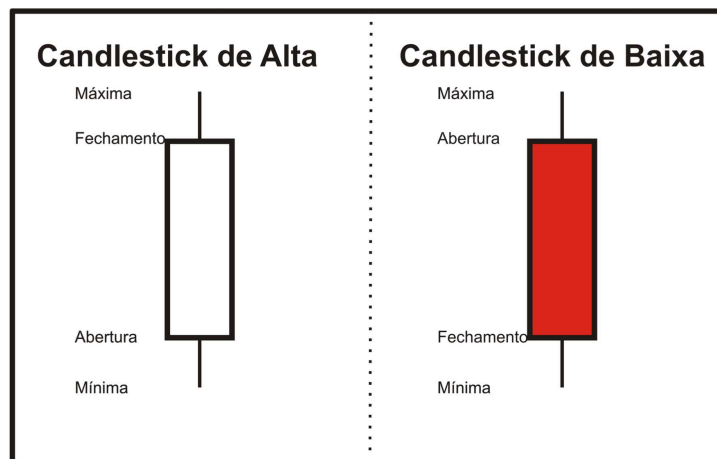


Figura 18: Candlestick

O padrão *Darkcloud* (figura 19) é formado por duas *candlesticks*, sendo a primeira de alta e a segunda de baixa, onde a segunda tem a abertura maior que a máxima da primeira e fechamento maior que a abertura da primeira (MATSURA, 2006, p.61). Bigalow faz a seguinte interpretação deste padrão.

Após uma forte tendência de alta, a atmosfera está altista. O otimismo está presente. O preço abre com *gap* para cima. Os vendedores aparecem e empurram os preços de volta para baixo. Finalmente, o preço fecha na ou próximo das mínimas do dia. O fechamento não confirmou a maioria dos ganhos do dia anterior. Os comprados agora estão preocupados. Obviamente observam que a tendência de alta deve se encerrar. Este sinal proporciona uma boa venda a descoberto, com um estope acima da máxima do dia da vela preta (*candlestick* de baixa) .
(BIGALOW, 2010, p.47)

Diante deste cenário, criou-se a seguinte hipótese para nortear a atividade: “A implementação da prova de conceito orientada a Sistemas Multiagentes não será possível ou mesmo demandará muito esforço por parte do desenvolvedor, impossibilitando o seu uso no contexto em estudo neste TCC”. Iniciaram-se, assim, as outras etapas ilustradas na figura 17.

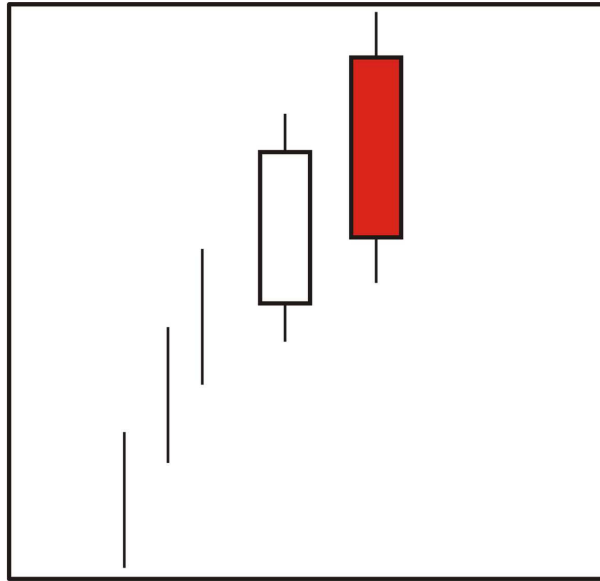


Figura 19: Padrão DarkCloud

Selecionar Ativos: Nesta etapa, foram obtidos os dados históricos dos ativos (i.e. Ações) das seguintes empresas: (i) Brookfield Incorporações, BISA3.SA; (ii) MVR Engenharia, MRVE3.SA; e (iii) Rossi Residencial S.A., RSID3.SA. Para realizar a atividade com os dados disponíveis aos produtos de Software escritos na linguagem Mql4 (KOVALYOV, 2000), foi feita uma coleta de dados do par de moedas Euro/Dólar americano (EUR/USD). Os códigos Mql4, Java+JADE e Python implementados neste estudo de caso estão disponíveis nos apêndices B, C e D, respectivamente. O período adotado para coleta foi o diário entre fevereiro de 2008 e janeiro de 2014.

Implementar Algoritmo: Considere a *candlestick* de baixa na figura 29 como *candlestick* 0, e a *candlestick* de alta como *candlestick* 1. De acordo com Matsura (2006,p.61), a formação é considerada Darkcloud se:

- A *candlestick* 1 deve ser de alta;
- A *candlestick* 0 deve ser de baixa;
- A *candlestick* 0 deve ter valor de abertura superior ao valor de fechamento da *candlestick* 1;
- A *candlestick* 0 deve ter seu valor de fechamento entre o valor mediano do corpo da *candlestick* 1. Deve também ter seu valor de fechamento superior ao valor de abertura da *candlestick* 1;
- A *candlestick* 0 deve ter um corpo considerável.

A Tabela 1 apresenta o tempo gasto para implementar cada algoritmo nas linguagens eleitas para este TCC.

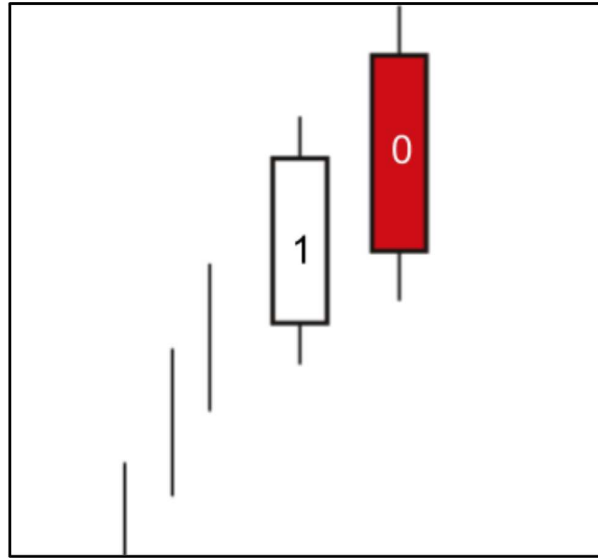


Figura 20: DarkCloud

Linguagem	Tempo médio
Python	2 Horas
Java + JADE	2 Horas
MQL4	2 Horas

Tabela 1: Tempo médio de implementação do algoritmo

O algoritmo foi implementado em Java combinado com JADE em um único comportamento, onde o algoritmo concentrado em uma classe é chamado pelo agente. Na linguagem Mql4, o algoritmo implementado retorna uma sinalização visual no gráfico de *candlestick*, fornecido pela plataforma, alertando a identificação do padrão. Na linguagem Python, o algoritmo implementado retorna uma listagem, contendo informações como datas de ocorrência do padrão, no terminal da IDE utilizada para implementação. Vale ressaltar, que a implementação foi o suficiente para estimar a viabilidade em se implementar a ferramenta proposta no Paradigma de Sistemas Multiagentes.

Analisar Resultado: Notou-se que o esforço necessário para implementar o algoritmo nas três linguagens foi aproximadamente o mesmo. Na linguagem Java+JADE e Python, por serem multiplataformas, permitiria desenvolver a ferramenta de forma que a mesma tenha o mesmo comportamento nas plataformas mais populares existentes hoje.

Na ferramenta desenvolvida, um critério de qualidade que merece atenção é a extensibilidade, visando futuras manutenções evolutivas bem como possibilitando trabalhar em mais de um contexto financeiro, como por exemplo, abordar o Mercado de Ações e o Mercado de Renda Fixa simultaneamente. Nesse caso, seria possível

trabalhar em um contexto de alto risco de investimento juntamente com um de baixo risco de investimento, obtendo uma melhor gestão de recursos investidos pelo usuário.

Em relação à extensibilidade no contexto financeiro, o Paradigma de Sistemas Multiagentes adequa-se ao estudo, visto que já existe um suporte tecnológico apropriado, um *framework* que atende às necessidades deste TCC. Esse *framework* oferece um suporte para troca de mensagens entre entidades inteligentes, no contexto abordado neste TCC, entre investidores. Tal suporte baseia-se na comunicação usando protocolos, padrões bem estabelecidos na comunidade de Software para Sistemas Multiagentes e possibilidades de extensibilidade orientadas, por exemplo, a ontologias específicas.

Outro ponto considerado foi que o esforço em se implementar a prova de conceito utilizando a combinação tecnológica linguagem Java e plataforma JADE foi aproximadamente o mesmo em relação às implementações em Python ou Mql4. Portanto, pode-se concluir que a hipótese inicial não se verificou. Assim, a aplicação do Paradigma de Sistemas Multiagentes no contexto financeiro abordado é possível, compreendendo um esforço por parte do desenvolvedor muito próximo aos exigidos nas demais linguagens analisadas.

Finalmente, diante das propriedades do Paradigma de Sistemas Multiagentes, expostas no **capítulo 2** e a descrição do contexto financeiro exposta no mesmo capítulo, nota-se que existem características comuns entre eles. Nesse cenário, merecem menção as seguintes características:

1. Autonomia - não há intervenções no processo de decisão de um agente, ele toma decisões sem intervenção de outros agentes ou humanos. Da mesma maneira, um investidor humano também possui autonomia para decidir quando realizar investimentos;
2. Sociabilidade - um agente interage com outros agentes, Software ou humano, através de algum tipo de linguagem de comunicação. Da mesma maneira, um investidor humano interage com outros atores presentes em uma bolsa de valores, seja para montar grupos, e de maneira conjunta resolver problemas que estão além das suas habilidades individuais, ou apenas para usar serviços disponibilizados por outros atores;
3. Reatividade - um agente é capaz de perceber o ambiente no qual está inserido através de seus sensores, é capaz de responder em tempo hábil às mudanças que ocorrem neste ambiente. Da mesma maneira, um investidor percebe o ambiente de bolsa de valores no qual está inserido, através de acompanhamentos frequentes de informações que alimentam suas estratégias, e responde às mu-

danças ocorridas na bolsa de valores de acordo com as informações de saída de suas estratégias;

4. Assincronismo - um agente, ao receber uma requisição, tem o poder de decidir se irá ou não atender à mesma. Adicionalmente, este agente pode enviar uma requisição a outro agente e continuar suas atividades. Caso não seja atendido, ele tem o poder de enviar a mesma requisição a outros agentes, por exemplo, contornando o problema. Da mesma maneira, um investidor humano pode procurar um ou vários atores presentes na bolsa e escolher aquele que melhor o atende.

Tal fato motivou a exploração do paradigma orientado a agentes no contexto financeiro abordado, cujo objetivo é contribuir com o contexto financeiro através da construção de uma ferramenta de estratégia financeira apoiada por Sistemas Multiagentes Comportamentais, onde não é exigido do usuário conhecimentos relacionados ao Mercado Financeiro. Assim, a ferramenta desenvolvida abstrai a complexidade de cálculos financeiros comumente utilizados na análise técnica e delega ao usuário somente a decisão de comprar ou vender.

4.5 Metodologia Adotada no Desenvolvimento da Ferramenta

A ferramenta resultante deste TCC teve seu desenvolvimento orientando-se de acordo com um modelo de desenvolvimento iterativo e incremental. Para isso, foi escolhida a metodologia ágil *Scrum* e foi feita uma adaptação desta, figura 21. As histórias de usuários foram divididas em duas categorias: (i) Produto, são as funcionalidades da ferramenta proposta; e (ii) Pesquisa, são pesquisas realizadas durante as atividades de Engenharia de Software Experimental, principalmente, em relação às provas de conceito. A Tabela 2 apresenta o *Product Backlog* inicial deste estudo. Vale ressaltar que ele foi modificado durante o desenvolvimento, como previsto no *Scrum*.

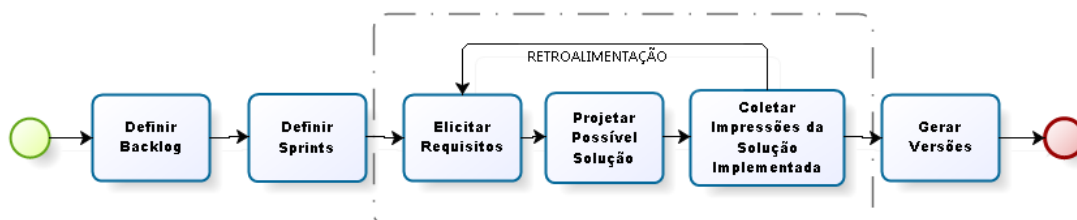


Figura 21: Scrum adaptado para o TCC

Tabela 2: Product Backlog inicial

Número	Descrição	Tipo
01	Como engenheiro, quero um agente capaz de montar e gerir uma carteira de ações de acordo com o perfil de investidor escolhido pelo usuário.	Produto
02	Como engenheiro, quero um agente capaz de tomar estratégias financeiras de acordo com o perfil de investidor escolhido pelo usuário e valor investido.	Produto
03	Como engenheiro, quero um agente capaz de realizar simulações de suas estratégias financeiras no período em que a Bolsa de Valores de São Paulo está fechada.	Produto
04	Como engenheiro, quero um agente capaz de realizar buscas contínuas de ações de empresas com presença na Bolsa de Valores de São Paulo e as categorize de acordo com seu setor e retorno diário.	Produto
05	Como engenheiro, quero um mecanismo no qual o usuário possa se cadastrar e se vincular a um grupo de agentes.	Produto
06	Como engenheiro, quero um código em Python capaz de reconhecer o padrão de <i>candlestick DarkCloud</i> , independentemente do ativo analisado para construir estratégias de investimentos mais eficientes.	Pesquisa
07	Como engenheiro, desejo definir estratégias de investimentos para servir de apoio aos agentes..	Pesquisa
08	Como engenheiro, quero um código em MQL4 capaz de reconhecer o padrão de <i>candlestick DarkCloud</i> , independentemente do ativo analisado para construir estratégias de investimentos mais eficientes .	Pesquisa
09	Como engenheiro, quero um código em Java+JADE capaz de reconhecer o padrão de <i>candlestick DarkCloud</i> , independentemente do ativo analisado para construir estratégias de investimentos mais eficientes.	Pesquisa

Nas estórias do tipo Produto, foram criadas tarefas que nortearam a implementação das estórias iniciais. Estas foram distribuídas em *Sprints* de 4 semanas bem como intercaladas com atividades de documentação de TCC. A listagem a seguir apresenta as tarefas por estória de usuário:

1. Eu como engenheiro, quero um agente capaz de montar e gerir uma carteira de ações de acordo com o perfil de investidor escolhido pelo usuário.
 - Implementar comunicação entre agentes gestores e caçadores;
 - Implementar comunicação entre agentes gestores e especialistas;
 - Implementar rotina de cálculo de riscos de carteiras para agentes gestores;
 - Implementar critério de intervenção em operações que representem risco em desacordo com o perfil escolhido pelo usuário;
 - Implementar rotina de criação e destruição de agentes especialistas;
 - Implementar rotina de montagem de carteira de ações;
 - Implementar rotina de autorização de operações.
2. Eu como engenheiro, quero um agente capaz de tomar estratégias financeiras de acordo com o perfil de investidor escolhido pelo usuário e valor investido.
 - Implementar rotinas de acompanhamento de ações de empresas que compõem a carteira de investimentos;
 - Implementar rotinas de solicitação de autorização de operações.
 - Implementar estratégia financeira baseada em Médias Móveis;
 - Implementar estratégia financeira baseada em Padrões de Candlesticks;
3. Eu como engenheiro, quero um agente capaz de realizar simulações de suas estratégias financeiras no período em que a Bolsa de Valores de São Paulo está fechada;
 - Implementar rotina de simulação de estratégias em dados históricos;
 - Implementar rotina de persistência de resultados obtidos.
4. Eu como engenheiro, quero um agente capaz de realizar buscas contínuas de ações de empresas com presença na Bolsa de Valores de São Paulo e as categorizar de acordo com seu setor e retorno diário.
 - Implementar rotinas de busca de ações na Bolsa de Valores de São Paulo;
 - Implementar rotinas de triagem e persistência de ações;
 - Implementar rotinas de criação de agentes caçadores.
5. Eu como engenheiro, quero um mecanismo, no qual o usuário possa se cadastrar e se vincular a um grupo de agentes.
 - Implementar interface com o usuário e rotina de login;
 - Implementar rotina de instanciação de agentes gestores.

Como previsto na metodologia Ágil *Scrum*, durante o desenvolvimento da ferramenta foi necessário modificar o *Backlog* inicial. Dentre estas modificações temos : (i) a divisão da história 01 em duas; (ii) a remoção da história 03; e (iii) a criação de uma história relacionada à comunicação dos usuários com seus respectivos agentes.

Quanto à história 01 - "Eu como engenheiro, quero um agente capaz de montar e gerir uma carteira de ações de acordo com o perfil de investidor escolhido pelo usuário e valor investido.". Esta foi dividida nas seguintes histórias com suas respectivas tarefas:

Estória 01: Eu como engenheiro, quero um agente capaz de montar uma carteira de ações de acordo com o perfil escolhido pelo usuário.

- Implementar comunicação entre agentes gestores e caçadores;
- Implementar comunicação entre agentes gestores e especialistas;
- Implementar rotina de montagem de carteira de ações;

Estória 10: Eu como engenheiro, quero um agente capaz de gerir uma carteira de ações de acordo com o perfil escolhido pelo usuário.

- Implementar rotina de cálculo de riscos de carteiras para agentes gestores;
- Implementar critério de intervenção em operações que representem risco em desacordo com o perfil escolhido pelo usuário;
- Implementar rotina de criação e destruição de agentes especialistas;
- Implementar rotina de autorização de operações.

Quanto à história 03 - "Como engenheiro, quero um agente capaz de realizar simulações de suas estratégias financeiras no período em que a bolsa de valores de São Paulo está fechada.". Esta foi removida do *Backlog*. Durante o desenvolvimento da ferramenta, verificou-se que a implementação desta história comprometeria a entrega final. Tal aspecto foi verificado, uma vez que esta história demanda a criação de uma máquina de aprendizado baseada em inteligência artificial de modo a aproveitar ao máximo os dados obtidos com simulações automáticas. Esta máquina de aprendizado é uma forte candidata à uma futura manutenção evolutiva.

Quanto à história 11 - "Como engenheiro, quero um agente capaz de prover a comunicação entre os usuários e seus respectivos agentes.". Esta foi criada durante o desenvolvimento da ferramenta, onde verificou-se a necessidade de criar um agente responsável por monitorar novos cadastros de usuários e, assim, criar e vincular um agente. Adicionalmente, verificou-se a necessidade de monitorar ações de *log in* dos usuários. O agente implementado nesta história é apresentado como Agente Criador no capítulo 5. As tarefas relacionadas a essa história são apresentadas a seguir.

Estória 11: Como engenheiro, quero um agente capaz de prover a comunicação entre os usuários e seus respectivos agentes.

- Implementar comportamento de monitoramento de criação de novos usuários;
- Implementar comportamento de monitoramento de *log in* de usuários;
- Implementar rotina de criação de agentes gestores;
- Implementar comunicação entre agentes criadores e gestores;

A tabela 3 apresenta o cronograma seguido no desenvolvimento das estórias de usuário, vale ressaltar que as atividades executadas nos meses de abril e maio de 2015 são relacionadas à atividade "Coletar Impressões da ferramenta", descrita brevemente na subseção 4.3. As tabelas 4, 5 e 6 apresentam o *Backlog* final da ferramenta, bem como os percentuais de conclusão das tarefas por estórias e da ferramenta de maneira geral.

Tabela 3: Cronograma

Sprint	Atividade	Data início	Data fim
Sprint 1	Implementar estórias 06/07/08/09	03/03/14	28/03/14
Sprint 2	Implementar estória 05	04/08/14	29/08/14
Sprint 3	Implementar estória 11	01/09/14	26/09/14
Sprint 4	Implementar estória 01	01/10/14	30/10/14
Sprint 5	Implementar estória 02	03/11/14	28/11/14
Sprint 6	Implementar estória 04	01/12/14	12/12/14
Sprint 7	Implementar estória 10	02/03/15	27/03/15
Sprint 8	Coleta de impressões	01/04/14	30/04/15
Sprint 9	Coleta de impressões	04/05/14	29/04/15

Tabela 4: Backlog final: estórias de produto concluídas

Estória	Tarefa	Percentual
01	Implementar comunicação entre agentes gestores e caçadores;	100%
	Implementar comunicação entre agentes gestores e especialistas;	100%

Continuação na próxima página

Tabela 4 – Continuação da página anterior

Estória	Tarefa	Percentual
	Implementar rotina de montagem de carteira de ações;	100%
02	Implementar rotinas de acompanhamento de ações de empresas que compõem a carteira de investimentos;	100%
	Implementar rotinas de solicitação de autorização de operações;	100%
	Implementar estratégia financeira baseada em Médias Móveis;	100%
	Implementar estratégia financeira Padrões de Candlesticks;	100%
04	Implementar rotinas de busca de ações na Bolsa de Valores São Paulo;	100%
	Implementar rotinas de triagem e persistência de ações;	100%
	Implementar rotinas de criação de agentes caçadores;	100%
05	Implementar interface com o usuário e rotina de login;	100%
	Implementar rotina de instanciação de agentes gestores;	100%
10	Implementar rotina de cálculo de riscos de carteiras para agentes gestores;	100%
	Implementar critério de intervenção em operações que representem risco em desacordo com o perfil escolhido pelo usuário;	100%
	Implementar rotina de criação e destruição de agentes especialistas;	100%
	Implementar rotina de autorização de operações. ;	100%
11	Implementar comportamento de monitoramento de criação de novos usuários;	100%
	Implementar comportamento de monitoramento de <i>log in</i> de usuários;	100%
	Implementar comunicação entre agentes criadores e gestores;	100%

Continuação na próxima página

Tabela 4 – Continuação da página anterior

Estória	Tarefa	Percentual
	Implementar rotina de criação de agentes gestores;	100%
	Total	100,00%

Tabela 5: Percentual de estórias de pesquisas concluídas

Número	Descrição	Percentual
06	Eu, como pesquisador, quero um código em Python capaz de reconhecer o Padrão de Candlestick DarkCloud, independentemente do ativo analisado para construir estratégias de investimentos mais eficientes	100%
07	Eu, como pesquisador, desejo definir estratégias de investimentos de curto prazo em um agente visando testar sua capacidade em atuar no mercado;	100%
08	Eu, como pesquisador, quero um código em MQL4 capaz de reconhecer o Padrão de Candlestick DarkCloud, independentemente do ativo analisado para construir estratégias de investimentos mais eficientes;	100%
09	Eu, como pesquisador, quero um código em Java+ JADE capaz de reconhecer o Padrão de Candlestick DarkCloud, independentemente do ativo analisado para construir estratégias de investimentos mais eficientes.	100%
	Total Concluído.	100%

Tabela 6: Percentual geral de conclusão da ferramenta

Item	Percentual do item	Percentual relativo
Estória de Produto	100% .	100% de 84%

Continuação na próxima página

Tabela 6 – Continuação da página anterior

Item	Percentual do item	Percentual Relativo
Estória de Pesquisa	100%	100% de 16%.
	Total geral	100%

4.6 Metodologia Adotada na Análise dos Resultados Obtidos

O resultado obtido neste TCC foi avaliado de maneira quantitativa e qualitativa.

Quantitativa através da cobertura de código e métricas de qualidade de código, tais como : (i) Complexidade Ciclomática - CC, adotado para auxiliar no controle dos possíveis caminhos dos algoritmos; (ii) Coesão e Acoplamento - SC, adotado para mensurar o grau de reusabilidade dos pacotes projetados; e (iii) Herança - DIT, adotado para mensurar o grau de encapsulamento. Qualitativa através de questionários aplicados em potenciais usuários da ferramenta com intuito de mensurar o grau de aceitação do usuário, bem como verificar os objetivos adotados neste TCC. Outros dados quanto aos resultados obtidos são apresentados detalhadamente no capítulo 6.

4.7 Resumo do Capítulo

Elucidou-se neste capítulo, um breve levantamento acerca das principais metodologias de pesquisa científica encontradas na literatura, tais como: (i) pesquisa descritiva; (ii) pesquisa explicativa; e (iii) pesquisa exploratória. Esta última sendo a que mais se adequou ao tema coberto neste Trabalho de Conclusão de Curso.

Foi descrita ainda, a metodologia utilizada para condução do Trabalho de Conclusão de Curso como um todo. Adicionalmente, foram descritas as metodologias que conduziram o desenvolvimento da ferramenta em si e a análise dos resultados obtidos.

Na subseção 4.4, foram descritas as atividades que compõem a metodologia utilizada para condução do Trabalho de Conclusão de Curso como um todo. Dentre as atividades descritas, destacou-se a atividade de desenvolvimento de provas de conceito, a qual foi de suma importância para definir o suporte tecnológico adequado ao desenvolvimento da ferramenta.

Na subseção 4.5, foi apresentada a metodologia que conduziu o desenvolvimento da ferramenta em si. No caso, optou-se por uma adaptação da metodologia ágil Scrum.

Foi apresentado ainda nesta subseção, o Product Backlog inicial da ferramenta bem como suas modificações e distribuição no cronograma.

Por fim, na subseção 4.6, foi apresentada a metodologia que orientou a análise dos resultados obtidos. No caso, optou-se por avaliá-los de forma quantitativa, através do levantamento da cobertura de código e de métricas de qualidade de código. Adicionalmente, optou-se por avaliá-los de forma qualitativa, através da especificação de categorias bem como de questionários aplicados junto aos potenciais usuários da ferramenta.

5 Ferramenta Financeira

Esse capítulo apresenta em detalhes a ferramenta desenvolvida para o contexto do Mercado Financeiro de Bolsa de Valores usando uma abordagem Multiagentes. O capítulo está organizado em seções, como ilustrado na figura 22. Na seção 5.1, serão detalhados os diferentes níveis arquiteturais que compõem a estrutura geral da ferramenta. Salienta-se que o primeiro nível arquitetural corresponde à arquitetura base, a qual orientou a comunicação dos agentes, o ciclo de vida dos mesmos, protocolos de interação, suas criações, registros de serviços prestados e outros detalhes inerentes à programação multiagentes. A arquitetura base utilizada é a da Plataforma JADE. Essa foi apresentada no capítulo de Suporte Tecnológico, seção 3.2.3. Em um segundo momento, será apresentada a arquitetura da ferramenta em si, a qual foi organizada de acordo com o padrão arquitetural *Model-View-Controller* (MVC). Adicionalmente, será apresentada a arquitetura da máquina de raciocínio dos agentes, onde cada agente será descrito em detalhes bem como suas estratégias de ação no contexto financeiro. Esse último nível arquitetural foi quem mais demandou esforços nesse Trabalho de Conclusão de Curso, por representar a lógica de raciocínio dos agentes, e, portanto, o *core* da ferramenta. Por fim, é acordada uma abordagem evolutiva, visando facilitar a manutenção bem como a refatoração da ferramenta. Os códigos fontes Grails e Java da Ferramenta desenvolvida neste Trabalho de Conclusão de Curso estão disponíveis nos seguintes endereços: (i) <https://github.com/kverrna/TCC_Ferramenta_Grails>; e (ii) <https://github.com/RamonCruzSoftware/TCC_Ferramenta_Java>.

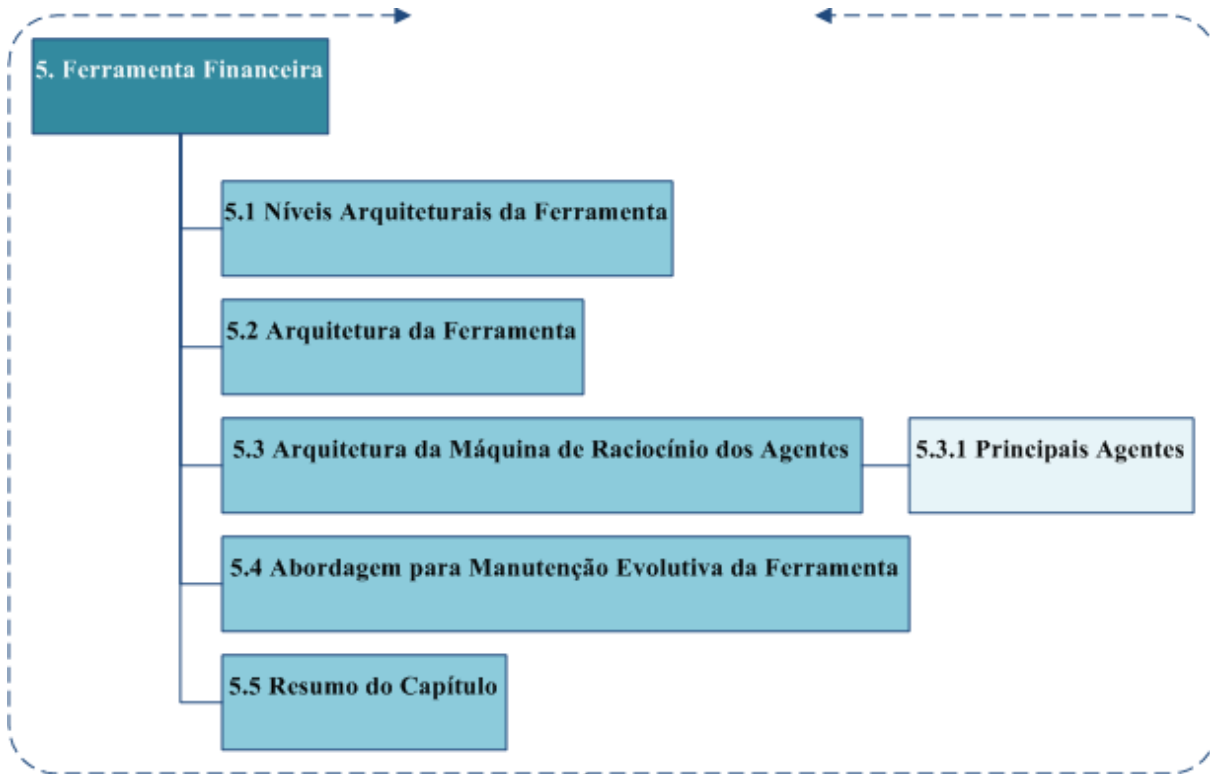


Figura 22: Organização do Capítulo

5.1 Níveis Arquiteturais da Ferramenta

A ferramenta compreende três níveis arquiteturais, sendo cada um deles brevemente descrito a seguir:

Primeiro Nível Arquitetural: esse primeiro nível arquitetural corresponde à arquitetura base. Nesse escopo, a ferramenta está orientada pela arquitetura da própria plataforma JADE. Essa arquitetura provê recursos, padronizados pela *Foundation for Intelligent Physical Agents* (FIPA) (TELECON, 2014), visando facilitar, dentre outras ações da programação multiagentes: (i) ciclo de vida dos agentes, orientado de acordo com uma máquina de estados desenhada especificamente para sistemas multiagentes comportamentais; (ii) comunicação entre os agentes, garantindo que a interação entre os agentes seja realizada via protocolos e ontologias pré-estabelecidos; (iii) registro dos serviços prestados pelos agentes, facilitando a busca por serviços bem como a localização de um agente em específico; (iv) criação dos agentes, (v) transporte de mensagens, com acompanhamento de troca de mensagens em tempo real, via ferramentas gráficas, e (vi) especificação dos comportamentos, orientando-se pelos tipos de comportamentos estabelecidos na plataforma, sejam eles simples ou compostos.

Segundo Nível Arquitetural: esse segundo nível arquitetural corresponde à arquite-

tura da ferramenta em si, a qual se orienta pelo padrão arquitetural MVC. Esse padrão foi utilizado, pois a ferramenta é disponibilizada via Web. Dessa forma, a estrutura de pacotes da ferramenta está organizada em três camadas: modelo, visão e controle. Detalhes desse nível arquitetural são acordados nesse capítulo, mais especificamente na próxima seção.

Terceiro Nível Arquitetural: esse nível arquitetural corresponde à arquitetura da máquina de raciocínio dos agentes. Essa pode ser vista como o core da ferramenta, pois seu escopo corresponde à lógica dos agentes. Essa lógica implementa o *rationale* dos agentes, ou seja, como eles raciocinam e quais ações desempenham. Esse nível arquitetural foi o que demandou maior dedicação ao longo do trabalho aqui apresentado. Visando facilitar o entendimento, a máquina de raciocínio dos agentes pode ser entendida como fazendo parte da camada de controle, quando associada ao modelo MVC. Esse nível arquitetural também é detalhado nesse capítulo, na seção 5.3.

5.2 Arquitetura da Ferramenta

Existem duas características que norteiam o desenvolvimento de um Software na atualidade, manutenibilidade e reusabilidade. Para viabilizar que a ferramenta desenvolvida neste TCC atenda essa visão de desenvolvimento de Software, manutenível bem como reutilizável, aliado ao fato da ferramenta ser *Web*, o modelo arquitetural escolhido foi o MVC (KRASNER; POPE, 1988). Dessa forma, os códigos implementados no Paradigma de Sistemas Multiagentes possuem um alto grau de modularização, baixo acoplamento e alta coesão. Assim, a ferramenta desenvolvida possibilita a realização de manutenções de maneira mais eficiente, dado que o contexto financeiro compreende regras de negócio de alta mutabilidade.

Em *View* ou Visão, figura 23, concentram-se todos os códigos relacionados à interface com o usuário, esta construída através do Grails versão 2.4.3. Em *Controller* ou Controle, figura 24, concentraram-se os códigos responsáveis por conectar o usuário com os agentes implementados, representando a máquina de raciocínio dos agentes - o *core* central do Sistema Multiagentes. Em *Model* ou Modelo, figura 24, concentram-se as entidades do modelo de domínio, representando os agentes de Software, outros modelos conceituais pertinentes (ex. protocolos de comunicação utilizados, regras de negócio do contexto financeiro e outros) e modelo de entidades da camada de persistência.

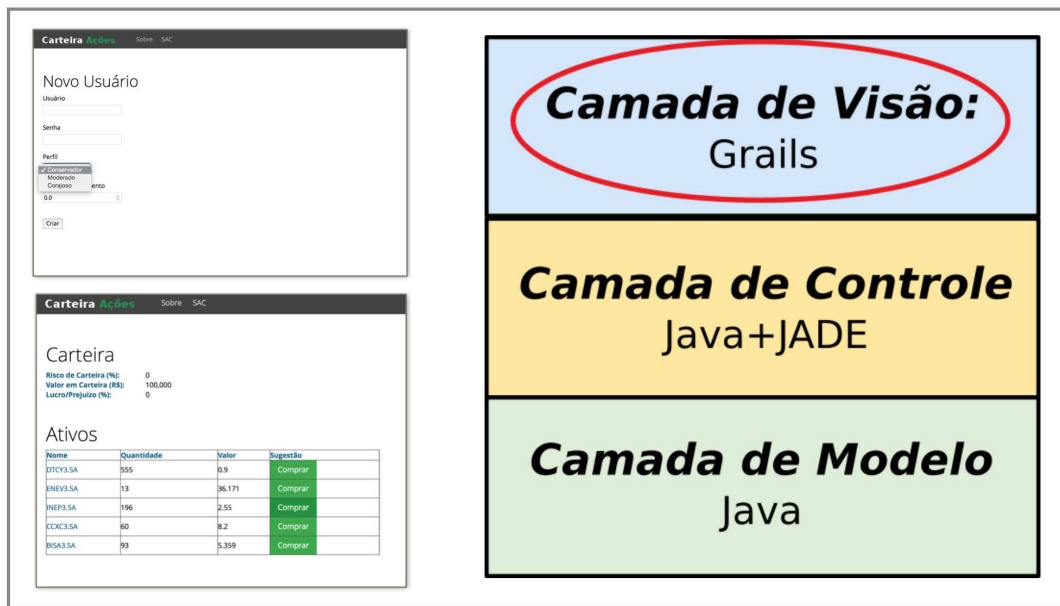


Figura 23: Camada View

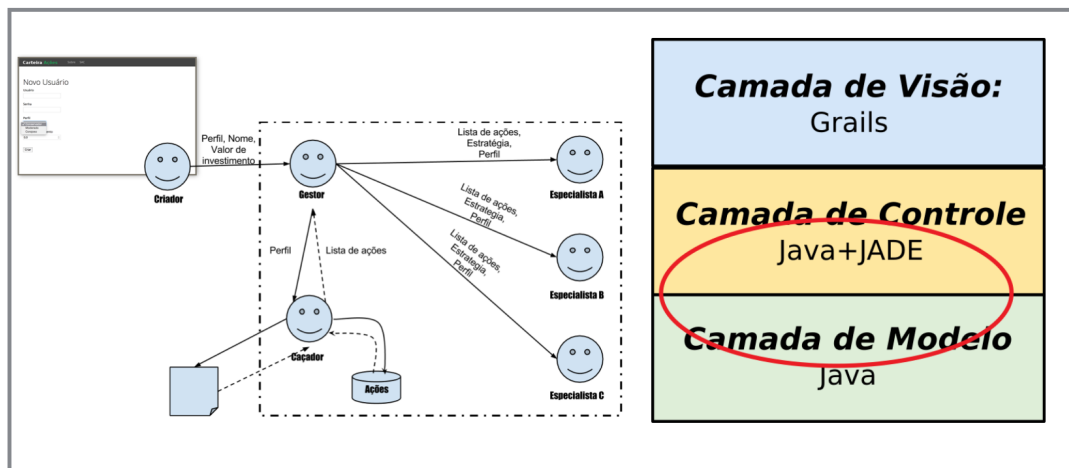


Figura 24: Controller-Model

Vale ressaltar que a Camada de Controle contém o pacote *core* (figura 25), nele estão concentrados os códigos que implementam os agentes comportamentais e suas classes de suporte. Apresenta-se neste pacote de maneira adicional, uma estrutura proposta para futura refatoração arquitetural da ferramenta (figura 37), em relação aos comportamentos dos agente.

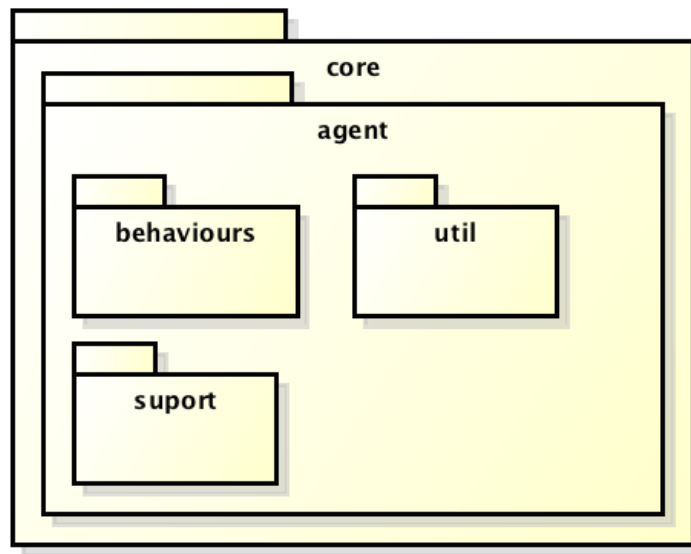


Figura 25: Pacote core

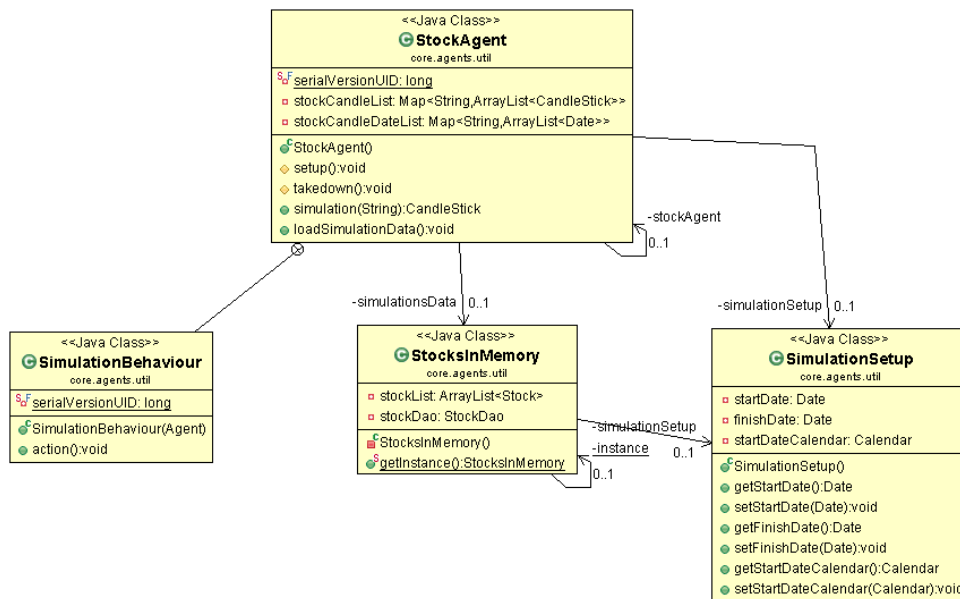


Figura 26: Estrutura elaborada para execução da Ferramenta em modo offline

O Subpacote *agent* contém os subpacotes: (i) *behaviours*, comporta classes que implementam comportamentos e estrutura de construção de comportamento proposto pelo autor como refatoração futura; (ii) *support*, comporta classes que auxiliam os agentes comportamentais implementados; e (iii) *util*, comporta classes que compõem a estrutura desenvolvida pelo autor para contornar o problema de ausência de conexão com internet.

A estrutura desenvolvida para contornar uma possível falha de ausência de conexão com internet é composta por quatro classes (figura 26): (i) *StockAgent*, classe que implementa um agente responsável por simular as ações do servidor de cotações; (ii) *Si-*

mulationBehaviour, classe que implementa o comportamento de resposta à uma requisição de cotação; (iii) *StocksInMemory*, classe utilizada para carregar uma seleção de dados na memória do computador, dados estes que serão enviados aos agentes que solicitarem cotações; e (iv) *SimulationSetup*, classe que contém dados essenciais para o funcionamento da ferramenta em modo *offline*.

A camada de Modelo comporta classes essenciais para o funcionamento da ferramenta e dos agentes em si. Estas classes estão organizadas no pacote *suport* (figura 27). O pacote *suport* contém três subpacotes: (i) *financial*, contém as classes responsáveis por prover estratégias financeiras a serem utilizadas por agentes especialistas (figura 28); (ii) *util*, contém classes responsáveis por prover o mecanismo de persistência em banco de dados e responsáveis pela lógica de requisição de dados de cotação de ações; e o subpacote (iii) *statistical*, contém as classes responsáveis por prover mecanismos de cálculos estatísticos utilizados pelos agentes caçadores, no processo de triagem de ações, e pelos agentes gestores, no processo de controle de risco.

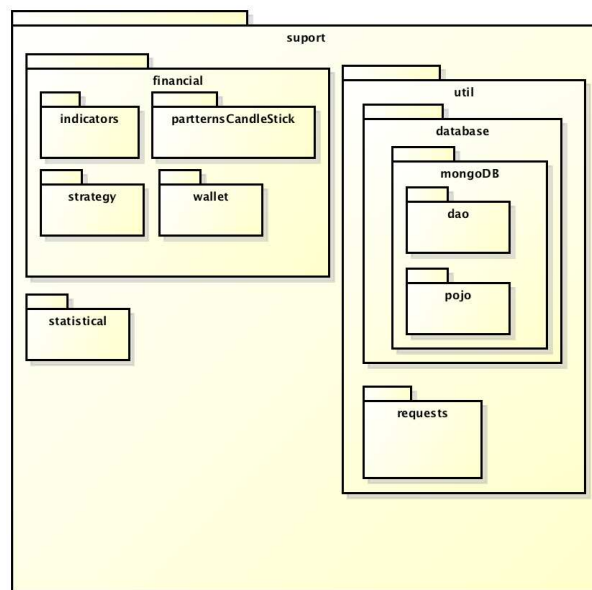


Figura 27: Pacote suport

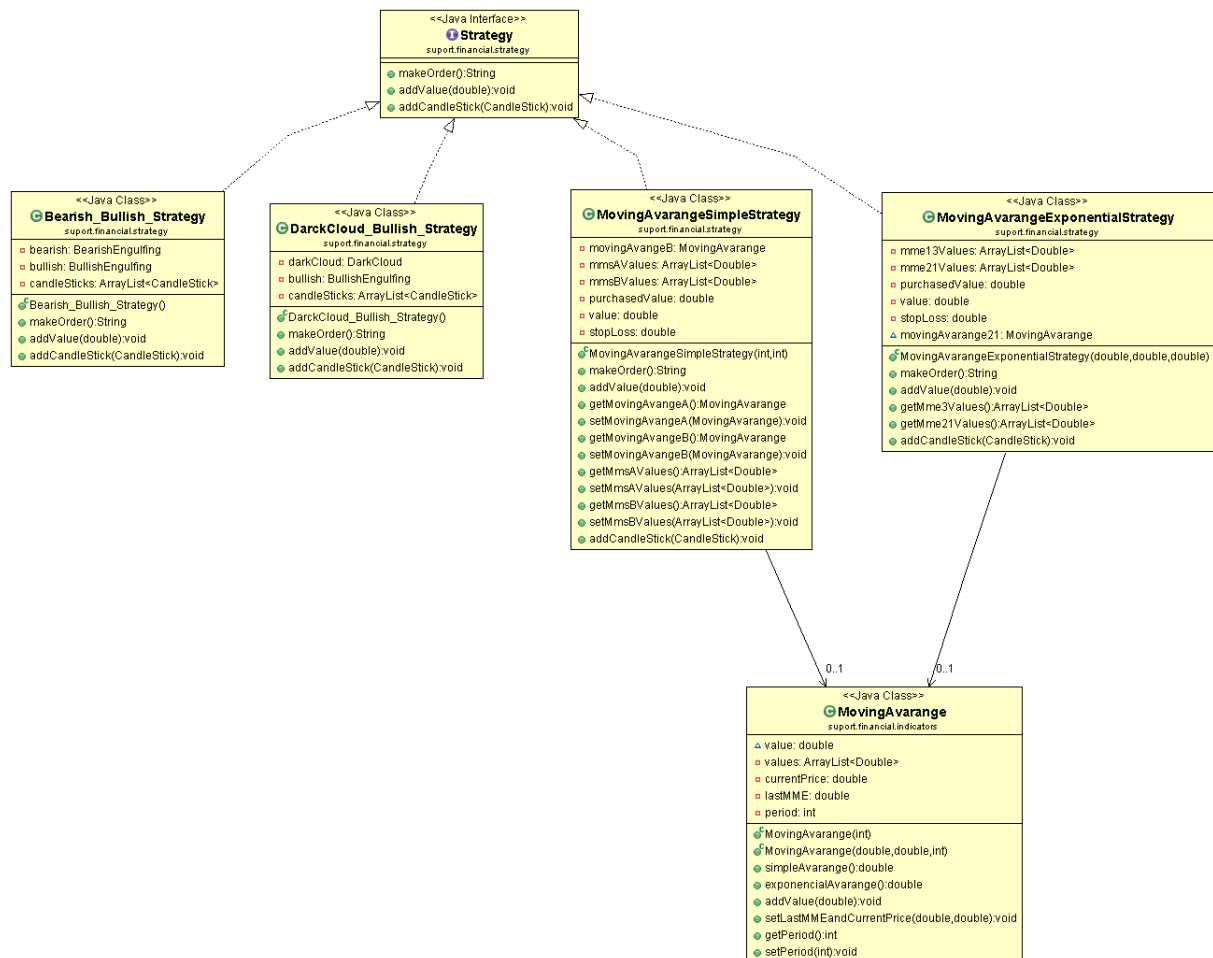


Figura 28: Estrutura desenvolvida para o uso de estratégias financeiras

5.3 Arquitetura da Máquina de Raciocínio dos Agentes

A ferramenta desenvolvida atua considerando três tipos de perfis de investidor, de forma a contribuir na realização do maior lucro possível de acordo com o perfil do usuário, ou seja, sob a perspectiva de investimento cabível para esse usuário. Informações sobre os perfis são simplificadaamente providas na Tabela 7.

Tabela 7: Perfis de investidores

Perfil	Descrição
Corajoso	São pessoas que, dentre os três perfis, assumem o maior risco e estão dispostas a acompanhar com maior frequência a movimentação do Mercado Financeiro.

Continuação na próxima página

Tabela 7 – Continuação da página anterior

Perfil	Descrição
Moderado	São pessoas que estão entre os perfis Corajoso e Conservador. Esse perfil adequa-se às pessoas que desejam operar esporadicamente no Mercado Financeiro, mas com um certo nível de risco.
Conservador	São pessoas que desejam gerir seu próprio plano de aposentadoria complementar, utilizando o Mercado Financeiro Brasileiro. Pessoas com este perfil desejam apenas arrecadar um capital suficiente para garantir seu padrão de vida, ou seja, são pessoas que desejam lucros razoáveis com um grau de risco menor do que os demais perfis.

5.3.1 Principais agentes

Foram projetados inicialmente três tipos de agentes comportamentais: (i) agente gestor; (ii) agente especialista, e (iii) agente caçador. No entanto, durante o desenvolvimento da ferramenta verificou-se a necessidade em criar um quarto tipo de agente de Software, o agente criador. Este é responsável por auxiliar na comunicação entre usuários *web* com seus respectivos agentes de Software, mais internos à lógica da ferramenta.

Agente gestor (*manager*): O agente do tipo gestor é o responsável por administrar a carteira de ações dos usuários, ações que por sua vez são acompanhadas por agentes especialistas. Ele autoriza uma compra ou venda de ações analisadas pelos agentes especialistas. Vale ressaltar que essa autorização passa pelo usuário vinculado ao grupo de agentes. O agente gestor, de maneira autônoma, faz o monitoramento de risco envolvido na carteira de ações administrada. Adicionalmente, executa o processo de montagem de carteira de maneira alinhada com o perfil escolhido pelo usuário. Suas responsabilidades são:

- Criar e excluir um ou mais agentes especialistas:

A quantidade de agentes especialistas criadas pelo agente gestor varia de acordo com o perfil de usuário. Portanto, podem ser criados: (i) dois agentes especi-

alistas para o perfil Corajoso; (ii) quatro agentes especialistas para o perfil Moderado; e (iii) sete agentes especialistas para o perfil Conservador.

- Montar carteira de ações:

Após instanciados os agentes para o usuário, o agente gestor verifica qual foi o perfil escolhido pelo usuário em seu cadastro e inicia o processo de montagem de carteira de ações (figura 24). Ele interage com um agente caçador e solicita um número determinado de ações, de acordo com o perfil do usuário. Segundo Pinheiro (2008, p.93), uma carteira com ações de 30 empresas diferentes pode ser considerada como uma carteira conservadora. Com base neste número, foi derivada a quantidade de ações por carteira de cada tipo de perfil. Assim, para o perfil Conservador, a carteira é formada por ações de 30 empresas. Para o perfil Moderado, 13 empresas. Já o perfil Corajoso, 8 empresas. Vale ressaltar que esses números iniciais foram baseados na recomendação do autor Pinheiro (2008, p.93), combinados com os números da sequência de Fibonacci. Essa sequência é bastante aceita no meio financeiro. Adicionalmente, esses valores iniciais poderão ser melhorados em futuras versões da ferramenta, caso seja necessário.

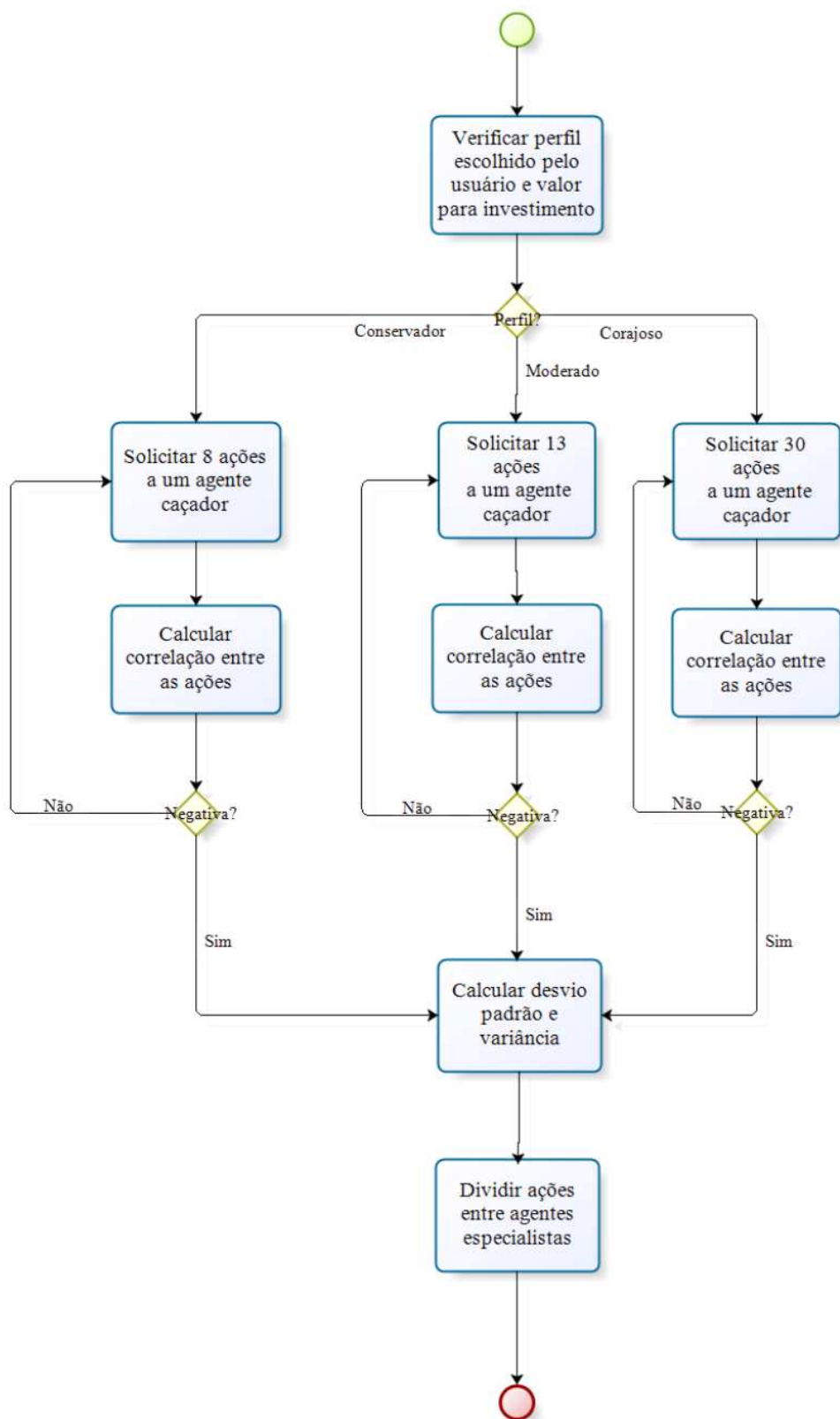


Figura 29: Processo de montagem de uma carteira

Ao receber o grupo de ações solicitadas ao agente caçador, o agente gestor calcula a correlação dos retornos (i.e. percentual de diferença entre o período atual com seu anterior) e as aprova de acordo com o perfil do usuário. Os critérios de aceite de uma ação são mostrados na Tabela 8.

Tabela 8: Critérios de aceite de ações

Perfil	Quantidade de empresas	Critério de aceite	Tolerância
Corajoso	8	Correlação negativa	Dois pares de ações com correlação positiva.
Moderado	13	Correlação negativa	Um par de ações com correlação positiva.
Conservador	30	Correlação negativa	Nenhuma ação.

Após feito cálculo da correlação, o agente gestor calcula o desvio padrão e a variância dos retornos da ação e distribui entre os agentes especialistas que compõem sua equipe.

- Intermediar compras ou vendas de ações:
Consiste em captar as sugestões de compra ou venda dos agentes especialistas e repassar ao usuário vinculado. Posteriormente, repassar aos agentes especialistas a autorização da compra ou venda de uma ação.
- Monitorar risco:
O Monitoramento consiste no cálculo constante do desvio padrão e a variância da carteira, bem como das ações individualmente. Vale ressaltar, que o risco é tratado nesta ferramenta de três maneiras combinadas. A primeira consiste no processo de montagem da carteira, onde são escolhidas ações de acordo com o perfil de usuário escolhido. A segunda consiste no *Stop Loss* presente nas estratégias utilizadas pelos agentes especialistas. E a terceira consiste no cálculo do desvio padrão e da variância realizado pelo agente gestor, como apresentado na figura 29.

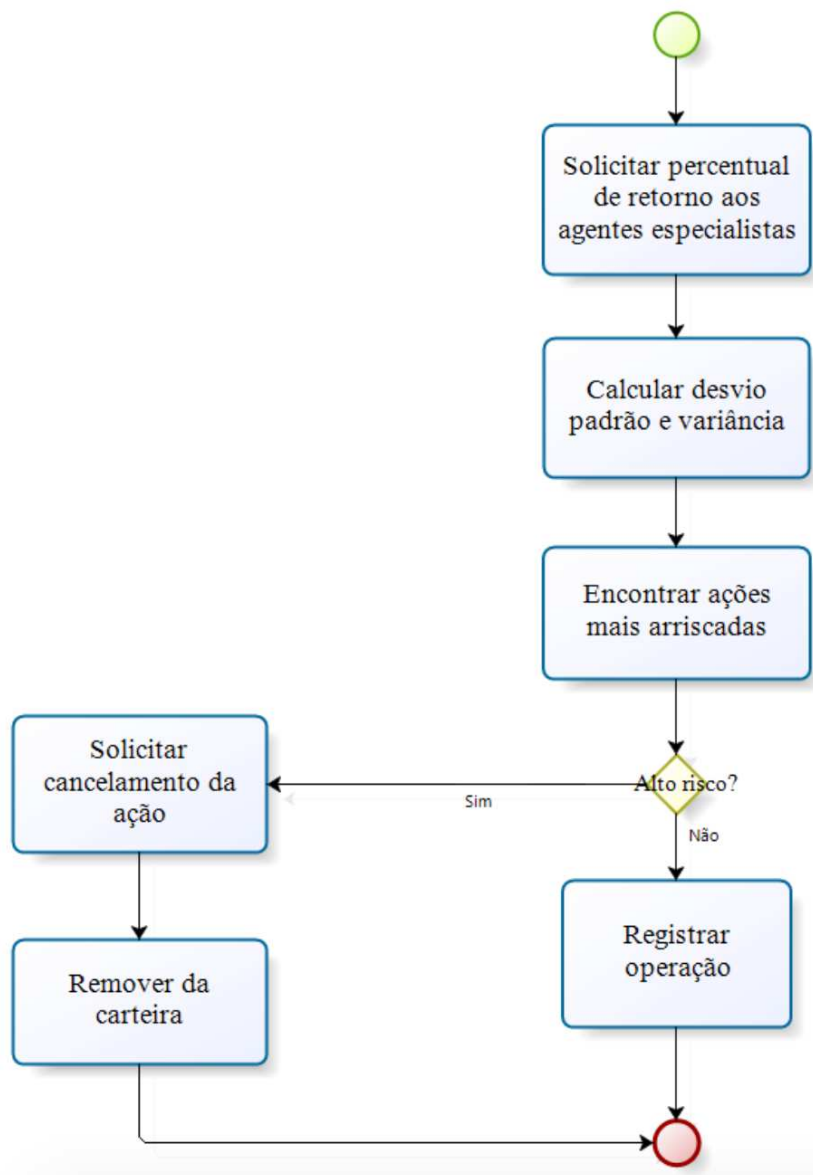


Figura 30: Processo de monitoramento de risco

Agente especialista (*expert*): O agente especialista é o agente responsável por realizar leituras constantes do mercado de valores em busca de novas cotações das ações sob sua responsabilidade. Ele é criado pelo seu agente gestor e executa estratégias compatíveis com o perfil escolhido pelo usuário. Recebe ainda um grupo de ações, ficando responsável pelas cotações das mesmas. Durante o desenvolvimento, verificou-se a necessidade de transferir a responsabilidade de mensurar riscos para o seu agente gestor. Suas responsabilidades são:

- Executar estratégias financeiras:

Consiste no uso das estratégias financeiras implementadas para esta ferramenta. É através de uma estratégia financeira que um agente especialista age

na Bolsa de Valores de São Paulo, ela é seu mecanismo de raciocínio. As estratégias financeiras adotadas para a ferramenta desenvolvida são compostas pelos seguintes indicadores financeiros:

1. Média Móvel Simples (MMS) e Médias Móveis Exponenciais (MME):

Médias móveis são médias de preços que se deslocam no tempo, ou seja, para um novo valor de entrada, há a saída de um valor antigo do cálculo. Com as médias móveis é possível suavizar a movimentação do mercado e, dessa forma, identificar tendências (MATSURA, 2006, p.68). Ao se calcular uma média móvel, é escolhida, primeiramente, qual a quantidade de valores que será utilizada no cálculo. Se utilizamos 5 valores, tem-se, então, uma média de 5 períodos, e assim por diante. Depois, define-se o tipo da média móvel. Para esta ferramenta, são as médias móveis simples e exponenciais. A Média Móvel Simples (figura 31) é uma média com tempo de reação mais lento dentre as médias. Ela é bastante utilizada para identificar tendências de alta ou de baixa no Mercado Financeiro. Quanto maior a sua periodicidade, mais lenta ela será. Médias de alta periodicidade, normalmente, são utilizadas em estratégias de longo prazo (MATSURA, 2006, p. 69).

$$MMS = \frac{\sum_1^i x_i}{i}$$

Figura 31: Média Móvel Simples

A Média Móvel Exponencial (figura 32) é mais ágil em relação à média simples. Ela confere peso maior aos valores mais recentes, e com isso se torna uma média mais sensível às oscilações do Mercado Financeiro. Na figura 32, a variável **F** representa o valor de fechamento de uma ação, **N** representa o número de períodos da média, e **K** representa a constante calculada.

$$MME = MME_{(x-1)} + K[F_x - MME_{(x-1)}]$$

$$K = \frac{2}{N + 1}$$

Figura 32: Média Móvel Exponencial

2. *Dark cloud*:

O padrão de formação *Dark cloud* (figura 33) é composto por duas candlesticks, a primeira de alta, em branco, e a segunda de baixa, em vermelho. Elas são precedidas de uma tendência de alta confirmada da ação. A segunda candlestick tem valor de abertura superior ao valor de fechamento da primeira, e tem valor de fechamento superior ao valor de abertura da primeira candlestick. Este padrão de formação indica reversão da tendência de alta para baixa (MATSURA, 2006, p.61). Tal fato será utilizado para sinalizar oportunidades de venda de ações.



Figura 33: Padrão DarkCloud

3. *Bearish Engulfing*:

O padrão de formação *Bearish Engulfing* (figura 34), assim como o *Dark cloud*, sinaliza a reversão de uma tendência de alta de uma ação. No entanto,

este padrão difere-se por sua segunda *candlestick* ter valor de fechamento inferior ao valor de abertura da primeira (BIGALOW, 2010, p. 38).



Figura 34: Padrão Bearish Engulfing

4. *Bullish Engulfing*:

O padrão de formação *Bullish Engulfing* (figura 34) é um padrão de reversão de tendência similar ao *Bearish Engulfing*. Este padrão é precedido de uma tendência de baixa, e sinaliza oportunidade de compra de ações. É composto por duas *candlesticks*, a primeira de baixa fecha de acordo com a tendência (i.e. uma *candlestick* de baixa). A segunda tem seu valor de abertura inferior ao valor de fechamento da primeira, e seu valor de fechamento maior do que o valor de abertura da primeira (BIGALOW, 2010, p. 36).



Figura 35: Padrão Bullish Engulfing

Adicionalmente, foram implementadas classes que representam estratégias financeiras que fazem uso de combinações destes indicadores, como apresentado na Tabela 9.

Tabela 9: Estratégias por perfil

Perfil	Estratégia 1	Estratégia 2	Estratégia 3
Corajoso	MME (13/21)	Dark Cloud + Bullish Engulfing.	Bearish Engulfing + Bullish Engulfing
Moderado	MMS(13/21)	MME(13/21)	MME(13/21)
Conservador	MMS(13/21)	MMS(21/34)	MME(21/34)

- Solicitar autorização para realizar compra ou venda de ações:
Consiste na interação com o agente gestor quando há algum sinal de compra ou venda de ações. Quando o agente especialista detecta, através de sua estratégia financeira, um sinal de compra ou venda, este solicita ao agente gestor a autorização, que por sua vez envia para o usuário na forma de sugestão quando o mesmo realizar o *log in* na ferramenta.

Agente caçador (hunter): O agente caçador é responsável por fazer buscas constantes de cotações de ações presentes na Bolsa de Valores e trabalhadas pela ferramenta. Suas responsabilidades são:

- Baixar e atualizar o banco de dados:
Consiste em realizar *downloads* de cotações de ações continuamente e atualizar os dados estatísticos no banco de dados.
- Categorizar ações utilizando critérios estatísticos:
Consiste em realizar uma categorização de ações através de valores estatísticos, tais como: (i) retorno médio diário; (ii) retorno médio 15 e 30 dias; (iii) variância média diária; e (iv) variância média 15 e 30 dias. Através destes valores, o agente caçador seleciona grupos de ações compatíveis com os perfis informados por agentes gestores no procedimento de montagem de carteira.
- Selecionar grupo de ações de acordo com o perfil de usuário:
Consiste em selecionar um grupo de ações usando como critério de seleção a variância combinada ao perfil de usuário.

Agente criador (*creator*): O agente criador é responsável por auxiliar a comunicação entre os usuários e seus respectivos grupos de agentes. Suas responsabilidades são:

Monitorar a criação de novas contas:

Consiste em monitorar o cadastro de usuários. Quando uma conta é criada, o agente criador captura as informações repassadas pelo usuário e instancia um agente gestor para este usuário. Em sequência, informa a esse agente gestor, através de uma mensagem com informações do usuário, tais como: nome, o valor investido e perfil escolhido.

Monitorar *Log In* de usuários:

Consiste em monitorar a autenticação de usuários na interface web. Quando um usuário autentica-se na ferramenta, o agente criador informa ao seu agente gestor para que o mesmo se comunique com seu usuário. A comunicação entre o agente gestor e seu respectivo usuário é demonstrado na figura 36, bem como o funcionamento geral da ferramenta.

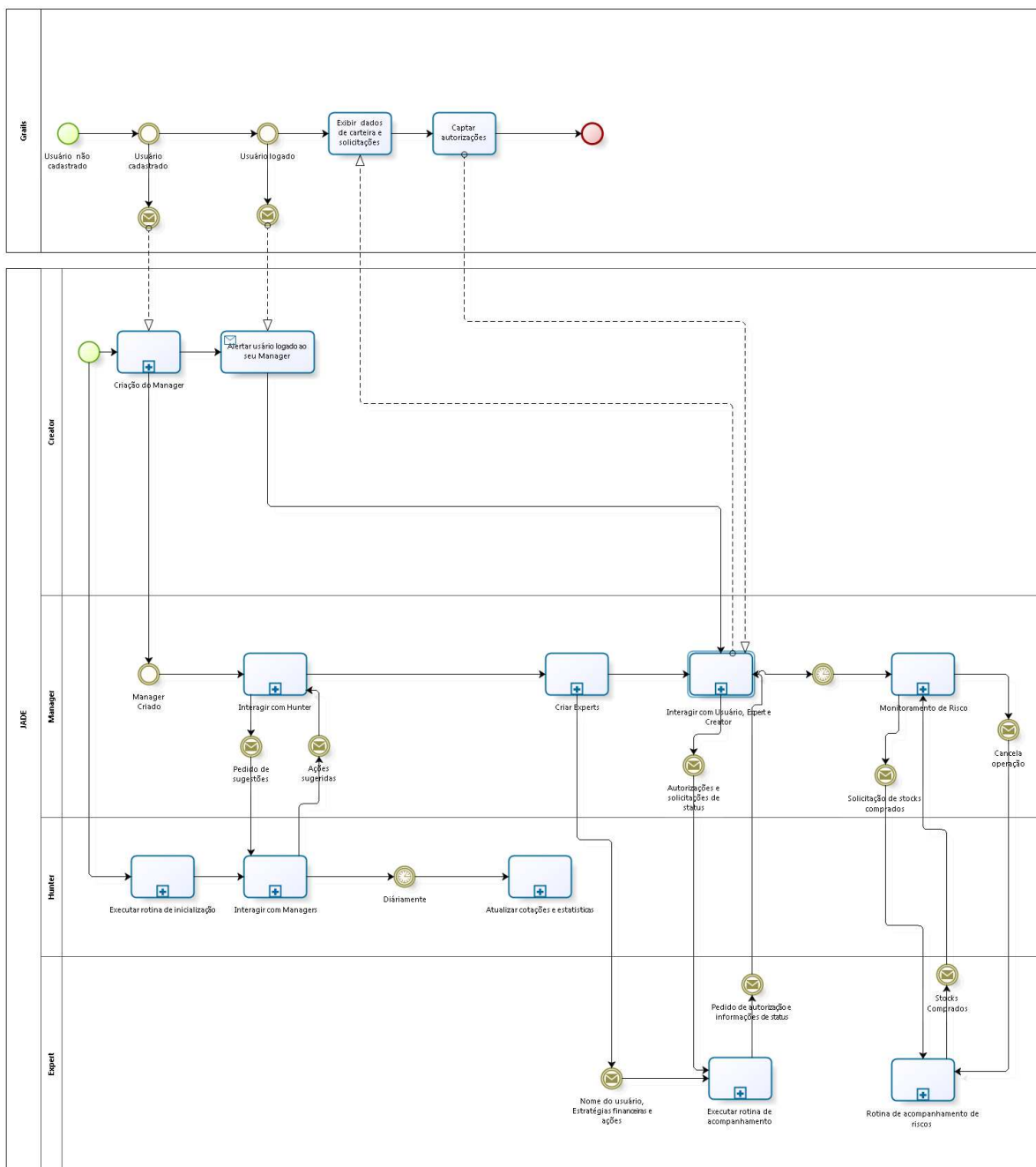


Figura 36: Visão geral da ferramenta

5.4 Abordagem para Manutenção Evolutiva da Ferramenta

A Ferramenta de Software desenvolvida foi desenhada de maneira a facilitar sua evolução. A figura 37 apresenta de maneira sucinta o uso de programação baseada em interfaces. O uso de interfaces, como orientado pelo padrão de projeto Strategy, permite a criação de novas estratégias financeiras, sejam estratégias baseadas em análise técnica ou análise fundamentalista. Este é um ponto importante para manutenções evolutivas em

relação ao contexto financeiro.

Em relação a Sistemas Multiagentes, durante o desenvolvimento da Ferramenta proposta e após seguir as práticas adotadas pelas literaturas relacionadas à implementação de agentes com o JADE, o desenvolvedor elaborou uma estrutura baseada em interfaces de maneira a reduzir o acoplamento de classes que implementam agentes comportamentais, bem como facilitar futuras manutenções. Esta estrutura contém uma interface denominada *ProcedureBehaviour* e uma classe concreta denominada *CommunicationBehaviour*, e é uma sugestão de refatoração para a Ferramenta de Software, figura 37.

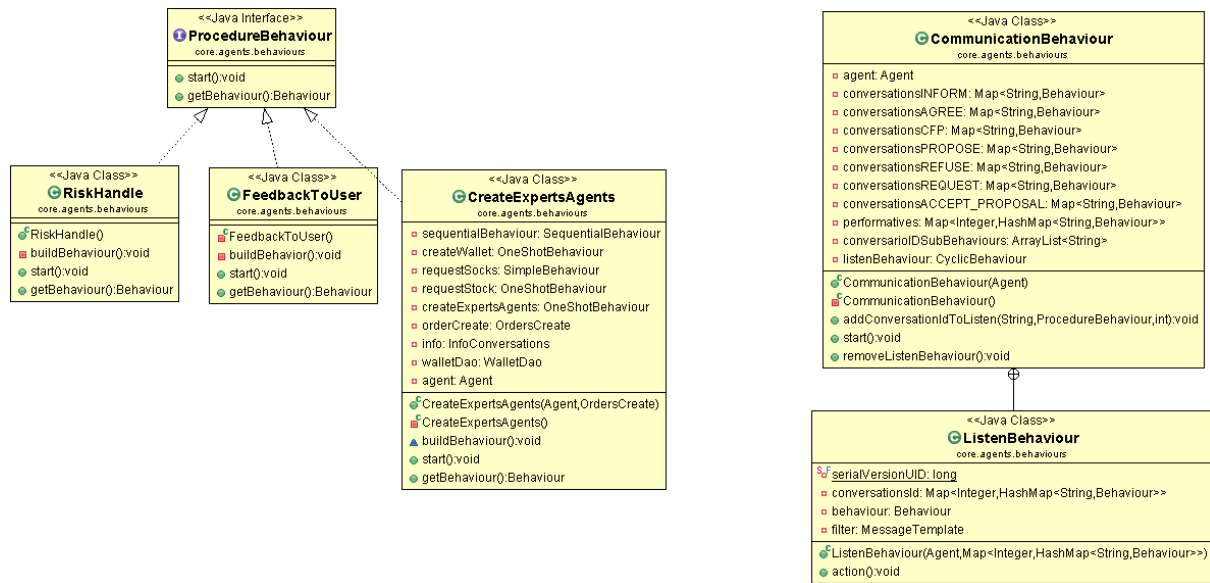


Figura 37: Estrutura sugerida para refatoração

A Classe *CommunicationBehaviour* implementa o comportamento em comum entre os agentes comportamentais, o qual consiste em receber mensagens de outros agentes e responder estas mensagens através da execução de um comportamento. Esta classe dispõe de um mecanismo de adição de comportamentos, no qual, para que essa adição seja feita, é necessário informar a classe, os atributos *ConversationID* e uma classe concreta *ProcedureBehaviour*. Esta classe agente pode utilizar a agregação para aplicar o seu conjunto de comportamentos, assim reduzindo o acoplamento da classe e aumentando sua coesão.

5.5 Resumo do Capítulo

Elucidou-se neste capítulo o detalhamento da arquitetura desenhada para a ferramenta. Essa arquitetura orientou-se de acordo com o padrão arquitetural *Model-View-Controller* (MVC). Adicionalmente, foi apresentada a arquitetura da máquina de raciocínio dos agentes implementados, esta composta por quatro tipos de agentes: (i) agente gestor, responsável por administrar uma carteira de ações; (ii) agente especialista, responsável identificar sinais de compra e venda de ações através de estratégia financeira; (iii)

agente caçador, responsável por baixar cotações e persistir dados estatísticos de ações; e (iv) agente criador, responsável por auxiliar a comunicação entre os usuários e seus respectivos agentes gestores. Ao final do capítulo, foi apresentada uma estrutura baseada em interfaces sugerida para facilitar a manutenção bem como a refatoração da ferramenta.

6 Resultados

Nesse capítulo, serão apresentados os principais resultados obtidos usando uma abordagem híbrida quantitativa e qualitativa, conduzida com pesquisa-ação combinada a cenários de uso. As subseções "Teste Unitários e Cobertura de Código" e "Métricas de Qualidade de Código" apresentam dados obtidos através de um levantamento quantitativo visando uma análise qualitativa do código fonte da ferramenta desenvolvida. A subseção "Pesquisa - Ação" apresenta dados relacionados às impressões dos usuários obtidas através de questionário a cada iteração da pesquisa ação, bem como os resultados obtidos com simulações das estratégias implementadas para a ferramenta.

A figura 38 ilustra a organização deste capítulo.

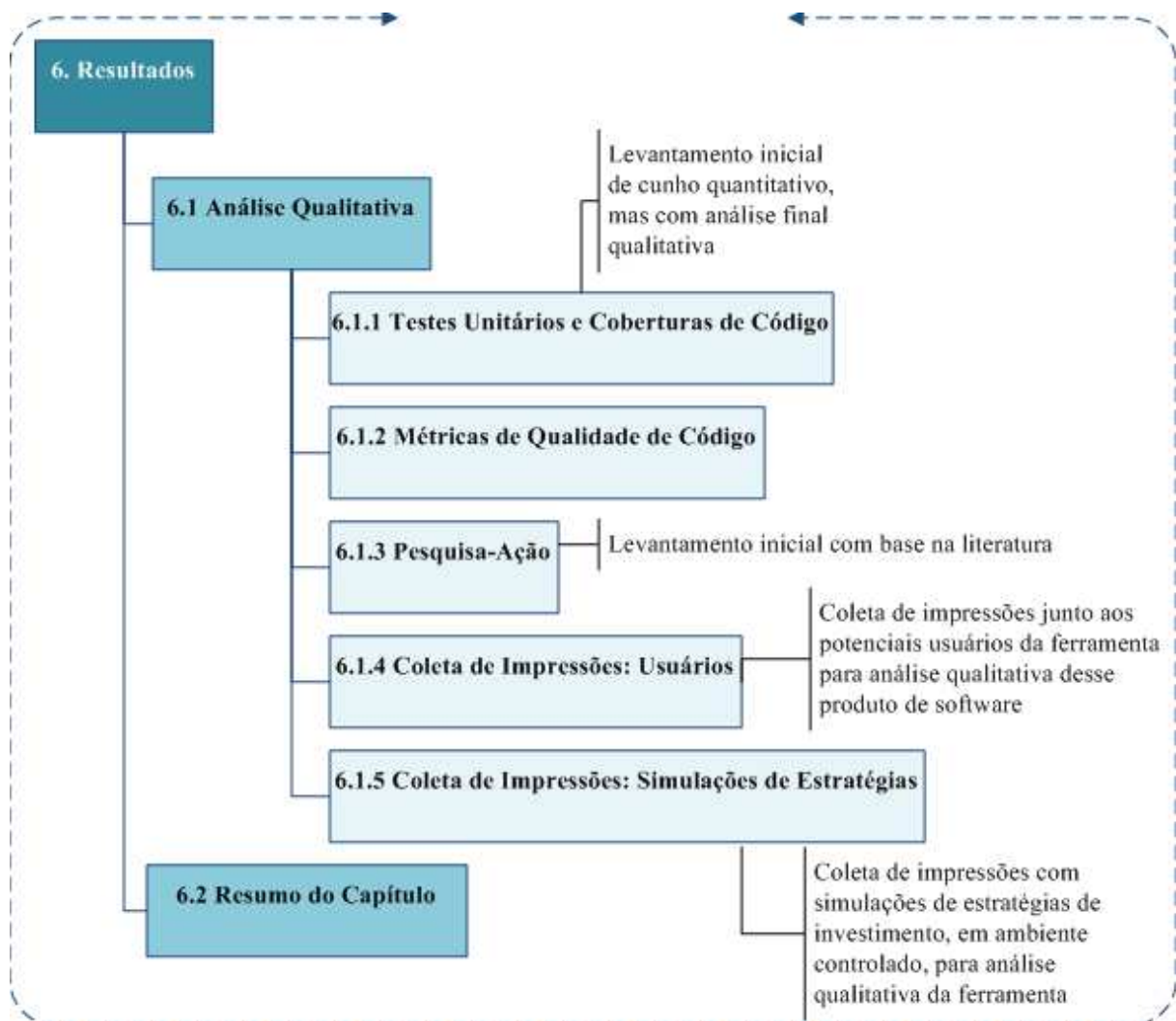


Figura 38: Organização do Capítulo

6.1 Análise Qualitativa

6.1.1 Teste Unitários e Coberturas de Código

Foram adotadas as ferramentas JUnit e Eclemma para a realização dos testes nos códigos referentes ao suporte, tais como: (i) os códigos que implementam a camada de persistência da ferramenta; (ii) os códigos que implementam algoritmos estatísticos utilizados pelos agentes caçadores e gestores; (iii) os códigos que implementam as estratégias financeiras utilizadas pelos agentes especialistas; e (iv) os códigos que implementam as classes de suporte a comportamentos utilizadas pelos agentes gestores e especialista. O propósito do uso dos teste unitários foi validar os algoritmos de suporte, este essenciais para o funcionamento da ferramenta. Assim, foi estabelecida a meta de 90% de cobertura de testes.

Quanto aos códigos que implementam os agentes comportamentais, estes não possuem, até o presente momento, uma ferramenta de validação comportamental adequada. Diante disto, foram migradas todas as rotinas utilizadas por comportamentos dos agentes, para classes que podem ser testadas através da ferramenta JUnit. A figura 39 apresenta a cobertura obtida da ferramenta em si, onde a meta de cobertura de código foi alcançada. Vale ressaltar, que o JUnit não é a ferramenta mais adequada para o teste de classes que possuem comportamentos utilizados por agentes. Assim, a meta de cobertura de testes unitários estabelecida para este Trabalho de Conclusão de Curso se aplica somente para as classes contidas no pacote *suport*.

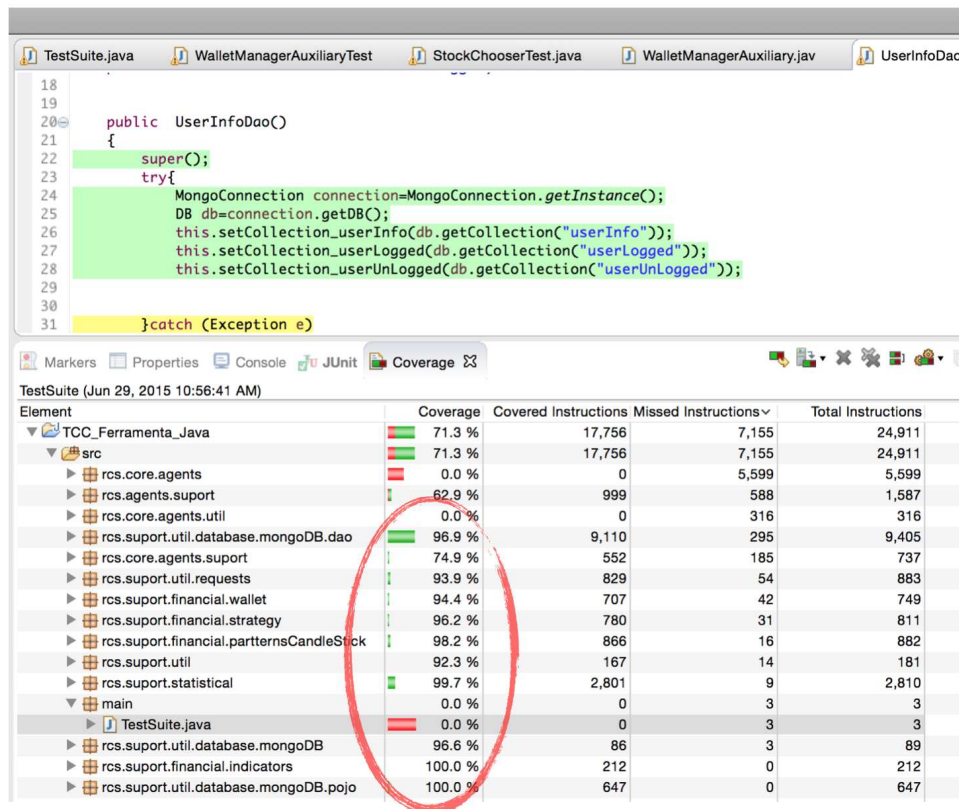


Figura 39: Cobertura de testes unitários

6.1.2 Métricas de Qualidade de Código

Segundo Filho (2013, p.1), medir e monitorar a qualidade do software é fundamental, independentemente da metodologia de desenvolvimento adotada. Ele afirma ainda que muitos fatores que compõem um bom software podem ser percebidos no código fonte. E mesmo que se possua uma ótima suite de testes, esta não é capaz de produzir informações acerca de itens de qualidade, tais como : (i) manutenibilidade; (ii) modularidade; (iii) flexibilidade e (iv) simplicidade. Assim, não é recomendado negligenciar as métricas de código em relação a outras abordagens de monitoramento da qualidade. Diante disso, as duas características fundamentais que nortearam o desenvolvimento da ferramenta resultado deste Trabalho de Conclusão de Curso foram, a manutenibilidade e a reusabilidade. Estas consideradas importantes pelo autor por contribuírem em termos de facilidade de evolução da ferramenta.

Assim, para verificar o grau de qualidade da ferramenta, no ponto de vista da qualidade de código, foram adotadas as seguintes métricas: (i) Complexidade Ciclomática - CC; (ii) Coesão e Acoplamento - SC; e (iii) Herança - DIT. Os valores de cada métrica foram obtidos com o auxílio da ferramenta Analizo. Esses valores são apresentados na tabela 10 e são classificados de acordo com os intervalos apresentados na figura 40.

	ACCM	SC	DIT
Excelente	[0 , 2]	[0 , 12 [[0 , 2 [
Bom	[2 , 7 [[12 , 28 [[2 , 4 [
Regular	[7 , 15 [[28 , 51 [[4 , 6 [
Preocupante	[15 , ∞ [[51 , ∞ [[6 , ∞ [

Figura 40: Intervalo para interpretação das métricas.

Fonte:([FILHO, 2013](#))

Tabela 10: Métricas de código fonte

Nome	Valor	Classificação
CC/ACCM	3.97	Bom
SC	12.07	Bom
DIT	0.36	Excelente

Por fim, diante dos dados apresentados na tabela 10, acredita-se que a ferramenta desenvolvida neste Trabalho de Conclusão de Curso possui uma arquitetura que contribui com a manutenibilidade e a reusabilidade.

6.1.3 Pesquisa - Ação

Segundo Rocha, pesquisa-ação pode ser considerada "independente" e "objetiva". E de acordo com Thiollent ([THIOLLENT, 1986](#) apud [ROCHA, 2012](#)), caracteriza-se pesquisa-ação como:

Um tipo de pesquisa social com base empírica, concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo no qual os pesquisadores e os participantes, representativos da situação e/ou do problema, estão envolvidos de forma cooperativa e participativa ([THIOLLENT, 1986](#) apud [ROCHA, 2012](#), p. 3).

A pesquisa-ação é um processo cíclico e contínuo, onde se planeja, implementa, descreve e avalia uma mudança para a melhoria de sua prática. Desta forma, há um aprendizado maior por parte do pesquisador durante o processo. Neste trabalho, o processo de pesquisa-ação foi utilizado para verificar o objetivo específico, sob o ponto de vista da interface com usuário, no caso: "Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta.". O cenário de uso bem como os resultados obtidos são apresentados nos tópicos seguintes. Em resumo, serão apresentadas as principais etapas para condução da coleta de impressões com os potenciais usuários da ferramenta. Posteriormente, serão descritos, na ordem: os objetivos dessa coleta, o público alvo avaliado, as observações coletadas e as ações tomadas.

6.1.4 Coleta de Impressões: Usuários

A coleta de dados foi organizada de acordo com as seguintes etapas:

- Planejamento:

Nesta etapa foram levantadas questões a serem respondidas pelos potenciais usuários de maneira a avaliar a interface gráfica da ferramenta;

- Implementação:

Nesta etapa foi elaborado um questionário (figura 41 e 42) a ser respondido pelos usuários, e foi disponibilizado ainda um cenário tecnológico onde a ferramenta apresentou dados em um período histórico de cotações da BM&FBovespa;

**FERRAMENTA FINANCEIRA APOIADA POR SISTEMAS MULTIAGENTES
COMPORTAMENTAIS**

A identificação no questionário é opcional.

Para as questões 1,2,4,5,6 e 7 use a escala apresentada na tabela abaixo.

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

Para a questão 3 considere os três perfis abaixo

Corajoso	Moderado	Conservador
----------	----------	-------------

1. Como você considera seu conhecimento sobre de Bolsa de Valores?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

2. Como você considera seus conhecimentos sobre de estratégias financeiras?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

3. Com qual dos três perfis disponíveis você se classifica?

Corajoso	Moderado	Conservador
----------	----------	-------------

4. Como você considera as cores da interface web da ferramenta?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

5. Como você considera a facilidade de compreensão da ferramenta?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

Figura 41: Questionário aplicado (PARTE I).

6. Como você considera a facilidade de uso da ferramenta?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

7. Como você considera a ferramenta no contexto geral?

Péssimo	Ruim	Razoável	Bom	Ótimo
---------	------	----------	-----	-------

8. Quais observações você tem para a ferramenta?

Figura 42: Questionário aplicado (PARTE II).

- **Descrição:**

Nesta etapa, concentra-se a descrição do cenário de uso da ferramenta. Quanto à descrição do cenário de uso, tem-se que o mesmo é apresentado a seguir:

A Ferramenta foi apresentada aos potenciais usuários em modo de simulação no período de 1 de março de 2013 a 1 de março de 2014, a versão do JADE utilizado é a 4.3.3 de 10 de dezembro de 2014, a versão do Grails foi a 2.4.3 e a versão do java foi a 1.7.0_79. A ferramenta foi simulada ainda em um notebook da marca Apple modelo MacBook Pro (Retina, 13-inch, Late 2013) com processador 2.6 GHz Intel Core i5, com 8 GB 1600 MHz DDR3 de Memória RAM e com o Sistema operacional OS X Yosemite versão 10.10.3.

- **Avaliação e Ação:**

Nesta etapa foram coletados os *feedbacks* dos potenciais usuários da ferramentas em primeiro momento; no segundo momento, foram tomadas ações de ajustes da interface da ferramenta, sendo apresentada novamente aos usuários. Dentre os *feedbacks* recebidos, destacou-se a sugestão de um usuário que se identificou com o perfil Corajoso, onde ele sugeriu usar outro indicador no lugar do indicador Média Móvel. As informações coletadas são expostas de maneira sucinta neste tópico e na íntegra no Apêndice E.

Objetivos: Esta coleta de dados com 12 potenciais usuários, com idade entre 18 e 26 anos, teve como principal propósito verificar o objetivo específico: “Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta”. Vale ressaltar que essa avaliação teve como foco a interface gráfica da ferramenta com os usuários, onde cada usuário recebeu um formulário com questões objetivas e um campo livre para sugestões. O comportamento dos agentes bem como os resultados das estratégias abordadas foram avaliados através de simulações.

Público alvo avaliado: A avaliação ocorreu em maio de 2014 e participaram da avaliação seis estudantes universitários com idade até 26 anos e com pouco conhecimento relacionado à estratégias financeiras, onde: 1 se classifica no perfil Corajoso; 1 se classifica no perfil Moderado e 4 se classificam no perfil Conservador. O teste funcional da ferramenta foi feito com uso de dados históricos no período entre 1 de março de 2014 e 3 de março de 2015. A interface gráfica da ferramenta foi o foco dessa primeira análise.

Observações coletadas: Quando questionados sobre as cores adotadas na ferramenta, 5 pessoas consideraram boas e 1 pessoa considerou ótima. Quando questionados

sobre a facilidade de compreensão da ferramenta, 3 pessoas consideraram como bom, 1 pessoa considerou razoável, 1 pessoa considerou como ruim e 1 se absteve. Quando solicitadas observações gerais para melhoria da ferramenta, destacaram-se: (i) feedback indicando o carregamento e o processo da ferramenta; (ii) atualização da página principal automaticamente; (iii) disponibilidade de um canal de dúvidas; (iv) não adoção de estratégias baseadas em Médias Móveis; (v) utilização de javascript na interface; (vi) execução da ferramenta em duas máquinas, e (vii) disponibilidade de informações sobre estratégias e uso da ferramenta.

As sugestões apresentadas pelos usuários demandaram melhorias na interface gráfica, como esperado, e uma demanda na melhoria das estratégias adotadas. Para atender a demanda de melhorias na interface gráfica, foi necessário realizar pequenas capacitações em tecnologias *front-end*, tais como: `JavaScript` e `jQuery`. Para atender a demanda de melhorias nas estratégias, foi necessário criar novas estratégias baseadas em outros indicadores financeiros e simular novamente a estratégia.

Ações Tomadas: Foram feitos ajustes na interface, dentre eles: (i) a criação um canal de comunicação por onde os usuários podem enviar dúvidas, sugestões entre outros; (ii) a criação de uma página com informações importantes da ferramenta; e (iii) a inserção de um mecanismo de requisição automática na página principal. Após as melhorias, a ferramenta foi apresentada novamente aos usuários e verificou-se que as melhorias foram aceitas e novos feedbacks foram coletados. Ao final do ciclo pesquisa-ção, foram registradas algumas demandas de manutenção evolutiva da ferramenta, tais como a inserção de estratégias baseadas em outros indicadores financeiros. Neste caso, optou-se pelo uso de padrões de projeto, como o `Strategy`, visando facilidades orientadas à manutenção, reutilização e extensibilidade. Detalhes sobre a abordagem de manutenção evolutiva utilizada encontram-se no capítulo 5, sobre a ferramenta.

6.1.5 Coleta de Impressões: Simulações de Estratégias

Esta coleta de dados foi organizada de acordo com as seguintes etapas:

- **Planejamento:**

Nesta etapa, foi definido um intervalo de tempo para simular as estratégias financeiras;

- **Execução:**

Nesta etapa, foram feitas simulações para cada perfil de usuário existente na ferramenta;

- **Avaliação:**

Nesta etapa, foram feitas interpretações dos dados coletados em primeiro momento;

Em segundo momento, foram feitos ajustes nas estratégias que apresentaram resultados não satisfatórios.

Objetivos: Esta coleta de impressões com potenciais usuários teve como principal propósito verificar o objetivo específico: *“Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta”*. Vale ressaltar que essa avaliação teve como foco as estratégias adotadas na ferramenta.

As estratégias foram simuladas no período de 1 de janeiro de 2012 a 1 de março de 2015, a versão do JADE utilizada é a 4.3.3 de 10 de dezembro de 2014, a versão do Grails foi a 2.4.3 e a versão do java foi a 1.7.0_79. A ferramenta foi simulada ainda em um notebook da marca Apple modelo MacBook Pro (Retina, 13-inch, Late 2013) com processador 2.6 GHz Intel Core i5, com 8 GB 1600 MHz DDR3 de Memória RAM e com o Sistema operacional OS X Yosemite versão 10.10.3.

Observações coletadas: O processo de coleta de impressões foi iniciado simulando o perfil Corajoso. Ao final, as estratégias se mostraram ineficientes. No período simulado, as estratégias apresentaram -18,08% de ganho, ou seja, houve uma perda de capital considerável e tal fato motivou um ajuste nas estratégias. Ao simular o perfil Moderado, as estratégias adotadas apresentaram 11,60% de ganho. Durante a simulação percebeu-se uma oportunidade de melhoria nas estratégias adotadas. Ao simular o perfil Conservador, as estratégias adotadas para este perfil apresentaram 7,8% de ganho. Apesar do ganho, outras melhorias nas estratégias foram motivadas. As tabelas 11, 12 e 13 apresentam os resultados para os perfis Conservador, Moderado e Corajoso.

Tabela 11: Estratégias Perfil Conservador e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Média Móvel Simples 13/21	-29,06%
A2	Media Móvel Simples 21/34	-31,74%
A3	Media Móvel Exponencial 21/34	34,96%
A4	Media Móvel Simples 13/21	4,93%
A5	Media Móvel Simples 21/34.	-60,41%
A6	Media Móvel Exponencial 21/34.	8,26%
A7	Media Móvel Simples 13/21.	59,91%
	TOTAL GERAL	7,8%

Tabela 12: Estratégias Perfil Moderado e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Media Móvel Exponencial 13/21	9,7%
A2	Media Móvel Simples 13/21	-24,36%
A3	Media Móvel Exponencial 13/21	41,89%
	TOTAL GERAL	11,60%

Tabela 13: Estratégias Perfil Corajoso e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Bearish - Bullish	-34,71%
A2	Dark Cloud - Bullish Engulf	-6,45%
	TOTAL GERAL	-18,08%

Ações tomadas As estratégias financeiras do perfil Corajoso, associadas aos agentes especialistas foram substituídas por estratégias baseadas em Médias Moveis Exponenciais com uso de *Stop Loss*. Para esta versão da Ferramenta de Software, não serão utilizadas as estratégias baseadas em padrões de Candlestick. Verificou-se, nesse caso, que para desenvolver um código capaz de identificar variações de um padrão de Candlestick, bem como validar os sinais, se falsos ou não, seria mais adequado usar uma abordagem de inteligência artificial orientada a mecanismos de aprendizagem. Tal mecanismo não é escopo deste trabalho, entretanto, trata-se de uma sugestão de melhoria para a ferramenta. Essa melhoria poderá usufruir da abordagem evolutiva adotada no desenvolvimento dessa ferramenta, ampliando ainda mais a capacidade de atuação dos agentes.

Tabela 14: Estratégias Perfil Corajoso e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Media Móvel Exponencial 13/21	33,00%
A2	Média Móvel Exponencial 13/21	72,47%
	TOTAL GERAL	60,27%

A tabela 14 apresenta os novos resultados da simulação do perfil Corajoso no mesmo período da simulação anterior. O valor do *Stop Loss* foi ajustado empiricamente, e percebeu-se que 5% é o valor ideal. Este valor de *Stop Loss* foi adotado para os três perfis existentes na Ferramenta. As tabelas 15 e 16 apresentam os resultados da simulação dos perfis Conservador e Moderado com suas respectivas estratégias ajustadas.

Tabela 15: Estratégias Perfil Moderado e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Media Móvel Exponencial 13/21	13,80%
A2	Media Móvel Simples 13/21	-3,85%
A3	Media Móvel Exponencial 13/21	82,45%
	TOTAL GERAL	34,77%

Tabela 16: Estratégias Perfil Conservador e Resultados

Agente	Estratégia	Percentual de Ganho
A1	Média Móvel Simples 13/21	-4,28%
A2	Media Móvel Simples 21/34	-8,75%
A3	Media Móvel Exponencial 21/34	64,81%
A4	Media Móvel Simples 13/21	50,33%
A5	Media Móvel Simples 21/34.	8,17%
A6	Media Móvel Exponencial 21/34.	25,91%

Continuação na próxima página

Tabela 16 – Continuação da página anterior

Agente	estratégia	Percentual de Ganho
A7	Media Móvel Simples 13/21.	116,76%
	TOTAL GERAL	38,58%

6.2 Resumo do capítulo

Este capítulo elucidou as análises dos resultados obtidos com os testes realizados na ferramenta. Quanto a análise quantitativa, foi estabelecida uma meta de cobertura de teste unitário em 90%. Vale ressaltar que essa meta se aplicou somente às classes contidas no pacote *support*. Foram extraídas três métricas de código, através do Software Analizo, que permitiram verificar a conformidade da ferramenta desenvolvida com o terceiro e quarto objetivo específico deste trabalho, "Procurar facilitar a reutilização de Software, em especial visando melhorar a manutenibilidade da ferramenta bem como sua extensibilidade em trabalhos futuros" e "Procurar reduzir o acoplamento e aumentar a coesão, apoiando-se em boas práticas da Orientação a Objetos". Os valores obtidos com a análise quantitativa serviram como subsídio para uma análise qualitativa da ferramenta em si.

Quanto à análise qualitativa, utilizou-se uma abordagem iterativa-evolutiva baseada em pesquisa-ação combinada a cenários de uso, cujo os objetivos foram verificar a conformidade da ferramenta com o primeiro objetivo específico deste Trabalho de Conclusão de Curso, que é "Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta".

7 Considerações Finais

Esse capítulo procura resgatar as principais contribuições oriundas desse Trabalho de Conclusão de Curso. O capítulo documenta algumas limitações da pesquisa realizada. Por fim, são apresentados possíveis pontos de extensibilidade da ferramenta, os quais poderão acordar evoluções e trabalhos futuros. A organização deste capítulo é ilustrada na figura 43.

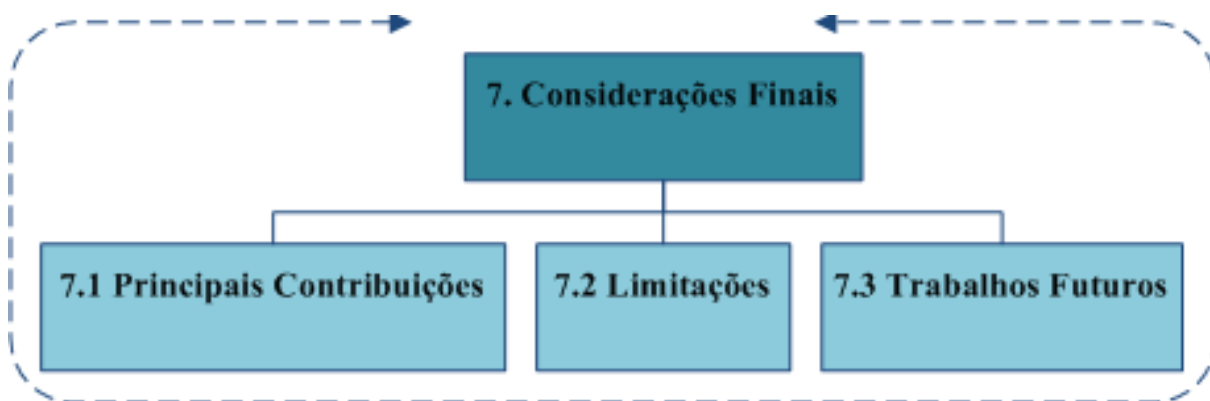


Figura 43: Organização do Capítulo

7.1 Principais Contribuições

A Ferramenta de Software desenvolvida neste TCC é capaz de auxiliar cidadãos brasileiros a realizarem investimentos em bolsa de valores. Sua primeira versão faz uso de estratégias comumente referenciadas por investidores, *Medias Móveis*. Entretanto, sua arquitetura foi desenhada de modo a suportar a adição de novas estratégias sem a necessidade de fazer alterações nos códigos fontes dos agentes comportamentais. Isto contribui para a manutenção evolutiva da ferramenta. Adicionalmente, a Ferramenta permite demonstrar o uso do paradigma de Sistemas Multiagentes como uma abstração válida para o contexto de bolsa de valores.

Ao usuário, não é requerido um prévio conhecimento relacionado ao contexto financeiro, sendo a ferramenta capaz de abstrair a complexidade de cálculos financeiros, deixando-os a cargo dos agentes. Nesse caso, ao usuário, é solicitado apenas a seleção do perfil (*Corajoso*, *Moderado* ou *Conservador*) e o valor de investimento. A Ferramenta, por ser *Web*, foi estruturada de acordo com o padrão arquitetural MVC, com a aplicação adicional de padrões de projeto. Em particular, o padrão *Strategy* destaca-se dentre os padrões de projeto adotados, uma vez que confere à ferramenta facilidades em termos de manutenção, refatoração, extensibilidade, e reusabilidade de código fonte.

Ressalta-se que os objetivos geral e específicos desse trabalho foram alcançados, de acordo com a análise dos resultados obtidos. Portanto, construiu-se, em atendimento ao objetivo geral, uma ferramenta de estratégia financeira apoiada por Sistemas Multiagentes Comportamentais, cuja máquina de raciocínio foi descrita em detalhes no capítulo 5 desse documento.

Procurou-se, em atendimento ao primeiro objetivo específico, trabalhar estratégias visando abstrair a complexidade dos cálculos financeiros comumente utilizados na análise técnica, deixando essa complexidade aos agentes e não aos usuários finais da ferramenta. Em atendimento ao segundo objetivo específico, explorou-se o Paradigma de Sistemas Multiagentes Comportamentais no contexto financeiro, usando provas de conceito pontuais e pesquisa-ação combinada a cenários de uso, conforme apresentado em detalhes no capítulo 6 desse documento. Em atendimento ao terceiro objetivo específico, desenvolveu-se uma arquitetura capaz de facilitar a Reutilização de Software, bem como a manutenibilidade e extensibilidade em trabalhos futuros. Por fim, em atendimento ao quarto e último objetivo específico, durante o desenvolvimento da Ferramenta adotaram-se boas práticas da Orientação a Objetos visando a redução do acoplamento e o aumento da coesão do código fonte.

7.2 Limitações

A ferramenta desenvolvida neste Trabalho de Conclusão de Curso aborda o contexto financeiro com estratégias oriundas da análise técnica, tais como: (i) estratégia baseada em médias móveis exponenciais e simples; e (ii) estratégias baseadas em padrões de formação de *candlestick*, inicialmente *Dark cloud* e *Bullish Engulf*. Ambas as estratégias são detalhadas no capítulo 5 deste trabalho.

Somente ações de empresas presentes na Bolsa de Valores de São Paulo são abordadas nesta versão da ferramenta. Entretanto, outros mercados pertencentes ao contexto financeiro poderiam ser vistos como pontos de extensão dessa pesquisa, tais como: (i) mercado de Renda Fixa; (ii) mercado de Opções; (iii) mercado de Contratos Futuros; (iv) mercado de Moedas (*FOREX*), entre outros. Ressalta-se que a ferramenta desenvolvida possui arquitetura que facilita sua evolução em termos de mercados financeiros, bem como as estratégias financeiras.

7.3 Sugestões de Trabalhos Futuros

Lembrando que a arquitetura adotada no desenvolvimento da ferramenta colabora com a manutenção evolutiva da mesma, poderão ser adicionados agentes capazes de atuar em outros contextos comumente encontrados na BM&FBovespa, tais como: (i) Mercado

de Opções; (i) Mercado de Mercadorias e Contratos Futuros; (iii) Renda Fixa, dentre outros. Adicionalmente, poderão ser incorporados agentes capazes de atuar em outras bolsas de valores em âmbito mundial, expandindo a capacidade cognitiva dos agentes com novas estratégias, novos indicadores financeiros e novas propostas de implementação (ex. implementação orientada ao Paradigma de Sistemas Multiagentes Intencionais, apoiado no Modelo BDI (*Belief-Desire-Intention*)).

Outro ponto de extensibilidade da ferramenta desenvolvida é a criação de uma interface *mobile* com o usuário, através do desenvolvimento de aplicativos para Smartphones e Tablets. Estes podem ser desenvolvidos para iOS, Android ou Windows Phone. Salienta-se que a interface gráfica da ferramenta desenvolvida, em sua versão atual, pode ser visualizada, via browser, em diferentes dispositivos móveis. Isso é possível, pois a mesma foi construída com base no framework Grails. Esse framework já confere, às aplicações desenvolvidas com base em sua arquitetura padrão, responsividade em diferentes dispositivos móveis. Entretanto, seria interessante desenvolver a ferramenta como uma aplicação nativa, específica para dispositivos móveis.

Referências

- ABREU, I. Falta de educação financeira. In: *Revista RI*. [S.l.: s.n.], ABRIL,2013. v. 172, p. 11. Citado na página 26.
- ALEXANDER, C. et al. A pattern language. Oxford University,New York, 1977. Citado na página 41.
- ALVARES, L. O.; SICHMAN, J. S. Introdução aos sistemas multiagentes. XVI JAI XVII Congresso da Sociedade Brasileira de Computação, Brasília-DF, 1997. Citado 2 vezes nas páginas 33 e 34.
- BIGALOW, S. W. *Operações Lucrativas com Candlestick: Identificando Oportunidades de mercador para Maximiza os Lucros*. [S.l.]: John Wiley & Sons,Inc., 2010. ISBN 978-0-470-92470-9. Citado 2 vezes nas páginas 54 e 81.
- BM&FBOVESPA. Balanço de operações de fevereiro de 2014. 2014. Disponível em: <<http://www.bmfbovespa.com.br/pt-br/a-bmfbovespa/sala-de-imprensa/Releases/2014/BMFBOVESPA-divulga-balanco-de-operacoes-de-fevereiro-2014-03-10.aspx?tipoNoticia=32&idioma=pt-br>>. Citado na página 25.
- BRATMAN. Intention, plans and practical reason. University of Chicago, 1999. Citado na página 34.
- BUFFET, M.; CLARK, D. *Warren Buffett e a análise de Balanços: Como identificar empresas com vantagem competitiva de longo prazo por meio de suas demonstrações financeiras*. 2. ed. Rio de Janeiro,Brasil: Sextante, 2010. Citado na página 31.
- CVM, C. d. V. M. Mercado de valores mobiliários brasileiro. Rio de Janeiro-RJ, v. 3, 2014. Citado 4 vezes nas páginas 29, 30, 31 e 36.
- FERBER; GASSER. Intelligence artificielle distribuee. Avignon, France, 1991. Citado na página 34.
- FILHO, C. M. de O. *Kalibro: interpretação de métricas de código-fonte*. Dissertação (Mestrado) — Universidade de São Paulo, São Paulo - SP, 2013. 89f. Citado 3 vezes nas páginas 16, 89 e 90.
- GAMMA, E. et al. *Padrões de Projetos : Soluções reutilizáveis de software orientado a objetos*. Porte Alegre,SC.: Bookman, 2000. ISBN 978-85-7307-610-3. Citado 3 vezes nas páginas 41, 42 e 43.
- GIL, A. C. *Como Elaborar Projetos de Pesquisa*. São Paulo: Atlas S.A, 2002. Citado 2 vezes nas páginas 49 e 50.
- GIRARDI. Agent-based application engineering. Setubal,Portugal, 2001. Citado na página 36.
- GIRARDI, R. Engenharia de software baseada em agentes. Universidade Federal do Maranhão, São Luís-MA, 2004. Disponível em: <<http://maae.deinf.ufma.br/Ensino/ES/CPGEE/Engenharia%20de%20Software%20baseada%20em%20Agentes.pdf>>. Citado 4 vezes nas páginas 32, 34, 35 e 36.

- HOLDOBLER; WILSON. The ants. Cambridge, MA, 1990. Citado na página 33.
- HYACINTH. Software agents: An overview. 1996. Citado na página 36.
- KNAPID; JOHNSON. Developing intelligent agents for distributed systems. New York, NY, 1998. Citado na página 35.
- KOVALYOV, S. *Programming in Algorithmic Language MQL4 - Introductory Course*. Londres: Cortez, 2000. Disponível em: <https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CDsQFjAC&url=http%3A%2F%2Fwww.vlkomarov.info%2Ffiles%2F2012%2F04%2Fmql4-book.doc&ei=fzaPVc_XBliZgwTQjYO4BQ&usg=AFQjCNFt2hT3vOYGGTXsViDAx9vRU-INJw&sig2=LFWL_wLriivCzWC39riBhA&bvm=bv.96783405,d.eXY>. Citado 2 vezes nas páginas 39 e 55.
- KRASNER, G. E.; POPE, S. T. A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. 1988. Disponível em: <<https://www.lri.fr/~mbl/ENS/FONDIHM/2013/papers/Krasner-JOOP88.pdf>>. Citado 3 vezes nas páginas 40, 41 e 69.
- MATSURA, E. *Comprar ou Vender? Como investir na Bolsa Utilizando Análise Gráfica*. 4. ed. São Paulo-SP: Editora Saraiva, 2006. ISBN 85-02-05911-4. Citado 5 vezes nas páginas 53, 54, 55, 79 e 80.
- METAQUOTES, S. c. *Linguagem MQL5 REFERENTE ao terminal do cliente MetaTrader 5*. Cortez, 2006. Disponível em: <<http://www.metaquotes.net>>. Citado na página 39.
- NORONHA, M. *É só Isso?! 1*. ed. Rio de Janeiro, Brasil: Imprinta Express Gráfica e Editora Ltda., 2010. ISBN 978-85-911077-0-4. Citado na página 31.
- OLIVEIRA, R. C. de. *Otimização da Confiabilidade via Sistemas Multiagentes baseado em colônia de formiga*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife-PE, 2008. 83f. Citado na página 33.
- ORACLE. Core j2ee patterns - data access object. 2015. Disponível em: <<http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>>. Citado na página 43.
- ORGE, P. C. A. *Como investir em ações: Aprenda a otimizar sua carteira de ações com estratégia usando MS Excel*. [S.l.]: Editora: Ciência Moderna, 2008. ISBN 978-8-573-93625-4. Citado na página 75.
- ROCHA, T. L. Viabilidade da utilização da pesquisa-ação em situações de ensino-aprendizagem. 2012. Disponível em: <http://www.unisc.br/portal/upload/com_arquivo/viabilidade_da_utilizacao_da_pesquisa_acao_em_situacoes_de_ensino_aprendizagem.pdf>. Citado 2 vezes nas páginas 28 e 90.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence A Modern Approach*. 2. ed. New Jersey, USA: Pearson Education, Inc., 2003. ISBN 0-13-790395-. Citado na página 32.
- SCHWABER, K.; SUTHERLAND, J. *VGuia do Scrum :Um guia definitivo para o Scrum-as regras do jogo*. [S.l.], 2013. 18 p. Disponível em: <<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-Portuguese-BR.pdf>>. Citado 2 vezes nas páginas 45 e 46.

- SERRANO, M. *Desenvolvimento Intencional de Software Transparente Baseado em Argumentação*. Dissertação (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro-RJ, 2011. 282f. Citado na página 34.
- TELECON. Java agent development framework. 2014. Disponível em: <<http://jade.tilab.com/>>. Citado 3 vezes nas páginas 26, 44 e 68.
- THIOLLENT, M. *Metodologia da pesquisa-ação*. São Paulo: Cortez, 1986. Citado na página 90.
- WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. 1. ed. London, England.: John Wiley & Sons, LTDA, 2002. Citado 2 vezes nas páginas 32 e 36.
- WOOLDRIDGE. Intelligent agents. MIT, MA, 1999. Citado 2 vezes nas páginas 34 e 36.

Apêndices

APÊNDICE A – Aposentadoria no Brasil

Aposentar-se é o ato de afastar-se de atividades profissionais. Após um período de contribuição em algum plano de aposentadoria, seja ele público ou privado, pode se ter um afastamento remunerado por tempo indeterminado. No Brasil, a aposentadoria é um direito garantido pela Constituição Federativa do Brasil (CF), no artigo 201 (BRASIL, 1988). Tem-se para tanto um Ministério responsável por amparar o trabalhador quanto aos seus direitos. Existe hoje no Brasil três Regimes Previdenciários: (i) o regime próprio (BRASIL, Constituição, 1988, artigo 40), este para servidores públicos, no qual o órgão em que ele está lotado possui um regime previdenciário; (ii) o regime geral (BRASIL, Constituição, 1988, artigo 201), este de natureza obrigatória e vínculo automático ao trabalhador de empresas privadas e públicas, sempre quando essas não possuem um regime previdenciário próprio; (iii) e o regime complementar (BRASIL, Constituição, 1988, artigo 202) (BRASIL, Emenda Constitucional nº 20, 1998), este de caráter facultativo. Pode ser do tipo fechado (BRASIL, Lei Complementar 108, 2001), ou seja, somente para trabalhadores que possuem vínculos empregatícios em uma determinada empresa. Mas, também pode ser do tipo aberto (BRASIL, Lei Complementar 109, 2001), ou seja, não vinculado ao trabalho, e normalmente gerido por empresas financeiras.

O regime geral de previdência é administrado pelo Instituto Nacional do Seguro Nacional, INSS. Trata-se de um Órgão vinculado ao Ministério da Previdência Social, criado em 24 de junho de 1990 (BRASIL, Decreto nº 99.350, 1990). Entretanto, o direito previdenciário existe desde 24 de janeiro de 1923, com o Decreto 4.682/23 (BRASIL, 1923). Os decretos supracitados garantem o direito à previdência ao trabalhador, denominado assegurado, e seus dependentes. São reconhecidos como dependentes (BRASIL, Lei nº 8.213, 1991, Art. 16): (i) o cônjuge, a companheira, o companheiro e o filho não emancipado, de qualquer condição, menor de 21 anos ou inválido ou que tenha deficiência intelectual que o torne absoluta ou relativamente incapaz, assim declarado judicialmente; (ii) os pais; (iii) e o irmão não emancipado, de qualquer condição, menor de 21 anos ou que tenha deficiência intelectual. No caso destes últimos, o INSS dispõe de pensão por morte e auxílio reclusão, quando o assegurado vinculado morre ou mesmo cumpre pena em regime carcerário.

Vale ressaltar ainda que todas as pessoas físicas brasileiras são asseguradas, as quais são divididas em segurados obrigatórios e facultativos (BRASIL, Lei nº 8.213, 1991 Art. 11). São denominados segurados obrigatórios: (i) todos os trabalhadores que prestam serviços de natureza urbana ou rural à empresa, em caráter não eventual, sob sua subordinação e mediante remuneração, inclusive como diretor empregado; (ii) empregados

domésticos, aqueles que prestam serviços de natureza contínua a uma pessoa ou família, no âmbito residencial desta, em atividades sem fins lucrativos; (iii) contribuinte individual, que são fazendeiros, pescadores e garimpeiros; (iv) trabalhadores avulsos, aqueles que prestam serviços para diversas empresas, sem vínculo empregatício, serviço de natureza urbana ou rural. São denominados segurados facultativos, todos os indivíduos maiores de 14 anos que se afiliam ao regime geral de previdência social, mediante contribuição, desde que não incluído nas disposições do artigo 11 (BRASIL, Lei nº 8.213, 1991, Art. 13).

Os segurados obrigatórios são vinculados de maneira automática ao INSS. Para estes são oferecidos oito benefícios (BRASIL, Lei nº 8.213, 1991, Art. 18), expostos abaixo:

- Aposentadorias

- Aposentadoria por idade: 65 para homens e 60 para mulheres;
- Aposentadoria por tempo de contribuição: 35 para homens e 30 mulheres;
- Aposentadoria por invalidez: em virtude de acidentes que invalide o segurado, de maneira permanente, à executar atividades profissionais;
- Aposentadoria especial: para profissionais que lidam com agentes químicos ou biológicos, os quais são considerados nocivos à sua saúde;

- Auxílios

- Auxílio doença: em períodos de doença, a qual impeça o segurado na realização de atividades profissionais de maneira temporária;
- Auxílio acidente: para acidentes ligados ao trabalho, que embora não tenha sido a causa única, tenha contribuído para morte do segurado, para redução ou perda da sua capacidade para o trabalho, ou produzido lesão que exija atenção médica para sua recuperação.

- Salários

- Salário maternidade: pago durante 120 dias, com início no período de 28 dias antes do parto e a data de ocorrência deste, observadas as situações e condições previstas na legislação no que concerne à proteção à maternidade;
- Salário família: pagos mensalmente ao segurado empregado, exceto ao doméstico, e ao segurado trabalhador avulso, na proporção do respectivo número de filhos ou equiparados nos termos do §2º da lei 8.213/91 (BRASIL, 1991).

Vale ressaltar que ao segurado que optar pela aposentadoria por tempo de contribuição, por exemplo, um homem que iniciou seu período de contribuição aos 18 anos

e completou seus 35 anos de contribuição aos 53 anos, este terá seu valor de benefício corrigido pelo fator previdenciário (BRASIL, Lei Nº 9.876, 1999). Nesse caso, calcula-se a média aritmética simples dos maiores salários-de-contribuição correspondentes a oitenta por cento de todo o período contributivo, multiplicada pelo fator previdenciário (BRASIL, Lei Nº 9.876, 1999). A fórmula do fator previdenciário é exposta na Figura seguinte.

Figura 2: Fator Previdenciário

Onde T_c representa o tempo de contribuição; A corresponde à alíquota, que é um valor estipulado pelo Ministério da Previdência Social (MINISTÉRIO DA PREVIDÊNCIA SOCIAL, 2014); E_s é a expectativa de sobrevida, calculada pelo IBGE, e I_d é a idade do segurado. Desta forma, o segurado que possui tempo de contribuição suficiente para requerer uma aposentadoria por tempo de contribuição não receberá o valor integral do benefício. O fator previdenciário é de uso obrigatório na aposentadoria por tempo de contribuição e facultativo na aposentadoria por idade.

Atualmente, o teto máximo oferecido pelo INSS é de R\$ 4.390,24 (MINISTÉRIO DA PREVIDÊNCIA SOCIAL, 2014), valor este reajustado anualmente. Ao assegurado que tem remuneração de valor até o teto máximo, a aposentadoria garantida por lei é suficiente para a manutenção do seu padrão de vida. Porém, ao segurado obrigatório, que possui remuneração superior ao teto máximo oferecido, o valor que será recebido a título de aposentadoria não será o suficiente para manter seus padrões. Estes últimos deverão procurar outros meios para que seja possível manter o seu nível de renda, sejam previdências privadas oferecidas por instituições financeiras ou investimentos feitos no mercado de capitais brasileiro e/ou estrangeiros. Se optar por realizar investimentos no mercado de capitais, este pode ser feito através de fundos de investimentos ou mesmo pessoalmente, apoiado por especialistas no assunto bem como utilizando ferramentas adequadas. Algumas dessas ferramentas são descritas na próxima subseção. O principal tipo de investidor que se pretende auxiliar com os estudos e suporte tecnológico providos nesse Trabalho de Conclusão de Curso é o investidor descrito neste parágrafo.

Referências

BRASIL. Constituição (1988). Constituição da República Federativa do Brasil. Brasília-DF, Senado Federal, 2014.

BRASIL, Decreto nº 99.350. Cria o Instituto Nacional do Seguro Nacional (INSS) define sua estrutura básica e o Quadro Distributivo de Cargos e Funções do Grupo-Direção e Assessoramento Superiores de suas Unidades Centrais e dá outras providências. Presidência da República. Brasília-DF, 1990. Disponível em

< http://www.planalto.gov.br/ccivil_03/decreto/Antigos/D99350.htm > acesso em

26 de maio de 2014.

BRASIL. Decreto 4.682 . Lei que cria, em cada uma das empresas de estradas de ferro existentes no país, uma caixa de aposentadoria e pensões para os respectivos empregados. Presidência da República. Rio de Janeiro-RJ, 1923. Disponível em <http://www.planalto.gov.br/ccivil_03/decreto/Historicos/DPL/DPL4682.htm > acesso em 26 de maio de 2014.

BRASIL, Lei da dos Benefícios da Previdência Social. Dispõe sobre a contribuição previdenciária do contribuinte individual, o cálculo do benefício, altera dispositivos das Leis nos 8.212 e 8.213, ambas de 24 de julho de 1991, e dá outras providências. Lei nº 9.876/91. Presidência da República, Casa Civil. Brasília-DF, 1991. Disponível em < http://www.planalto.gov.br/ccivil_03/leis/l8213cons.htm> acesso em 25 de maio de 2014.

BRASIL. Lei Complementar nº 108. Dispõe sobre a relação entre a União, os Estados, o Distrito Federal e os Municípios, suas autarquias, fundações, sociedades de economia mista e outras entidades públicas e suas respectivas entidades fechadas de previdência complementar, e dá outras providências. Presidência da República. Brasília-DF, 2001. Disponível em <http://www.planalto.gov.br/ccivil_03/leis/lcp/lcp108.htm> Acesso em 26 de maio de 2014.

BRASIL. Lei Complementar nº 109. Dispõe sobre o Regime de Previdência Complementar e dá outras providências. Presidência da República. Brasília-DF, 2001. Disponível em <http://www.planalto.gov.br/ccivil_03/leis/lcp/lcp109.htm> Acesso em 26 de maio de 2014.

MINISTERIO DA PREVIDENCIA SOCIAL. Fator Previdenciário. Brasília –DF. Disponível em <<http://www.previdencia.gov.br/informaes-2/fator-previdencirio-2/>> acesso em 25 de maio de 2014.

MINISTÉRIO DA PREVIDENCIA SOCIAL. Portaria Interministerial MPS/MF Nº 19, de 10 de Janeiro de 2014-DOU de 13/01/2014. Brasília-DF. Disponível em

<<http://www3.dataprev.gov.br/sislex/paginas/65/mf-mps/2014/19.htm>> acesso em 25 de maio de 2014.

APÊNDICE B – Algoritmo Dark Cloud em MQL4

```
#include
using namespace std;
int main()
{
    /* comentario */
    int n, i, a = 0, b = 1, F;
    cout << "Digite o numero de termos da sequencia de Fibonacci: ";
    cin >> n;
    cout << a << " " << b << " ";
    for (i = 0; i < n - 2; i++) {
        F = a + b;
        cout << F << " ";
        a = b;
        b = F;
    } cout << endl; return 0;
}
```


APÊNDICE C – Algoritmo Dark Cloud em Java + JADE

C.1 Classe DarkCloud

```

package patterns.CandleStick;

import java.util.ArrayList;

import candlesticks.CandleStick;

public class DarkCloud extends Pattern
{
    ArrayList<CandleStick> list;

    public DarkCloud ( ArrayList<CandleStick> list )
    {
        this.list = list;
    }

    public ArrayList<CandleStick> findCandleSticksPatterns ()
    {
        ArrayList<CandleStick> listResult = new ArrayList<CandleStick>();

        for (int i = 0; i < list.size(); i++)
        {
            if ((i + 1) < list.size())
            {
                if ((list.get(i + 1).getClose() > list.get(i + 1).getOpen()) &&
                    (((list.get(i + 1).getClose() + list.get(i + 1).getOpen()) / 2)
                     > list.get(i).getClose()) &&
                     (list.get(i).getOpen() > list.get(i).getClose()) &&
                     (list.get(i).getOpen() > list.get(i + 1).getClose()) &&
                     (list.get(i).getClose() > list.get(i + 1).getOpen()) &&
                     (((list.get(i).getOpen() - list.get(i).getClose()) /
                      (0.001 + list.get(i).getHigh() - list.get(i)

```

```

        .getLow()))
    > 0.6))

        {
listResult.add(list.get(i));
        }

    }

}

return listResult;
}

}

```

C.2 Classe abstrata Pattern

```

package patternsCandleStick;

import java.util.ArrayList;

import candlesticks.CandleStick;

public abstract class Pattern {

    public abstract ArrayList<CandleStick> findCandleSticksPatterns();

}

```

C.3 Classe CandleStick

```

package candlesticks;

public class CandleStick {

    private double open, high, low, close, volume;
    private String date;

    public CandleStick(double open, double high, double low,
        double close, double volume, String date)
    {

```

```
{
this.open = open;
this.high = high;
this.low = low;
this.close = close;
this.volume = volume;
this.date = date;
}

public CandleStick(String open, String high, String low, String close,
                  String volume, String date)
{
this.date = date;
this.open = Double.parseDouble(open);
this.high = Double.parseDouble(high);
this.low = Double.parseDouble(low);
this.close = Double.parseDouble(close);
this.volume = Double.parseDouble(volume);
}

public double getOpen()
{
    return open;
}

public double getHigh()
{
    return high;
}

public double getLow()
{
    return low;
}

public double getClose()
{
    return close;
}
```

```
public double getVolume ()
{
    return volume;
}

public String getDate ()
{
    return date;
}

public String getInformation ()
{
    return "Date: " + this.date +
        " _Open: " + this.open +
        " _High: " + this.high +
        " _Low: " + this.low +
        " _Close: " + this.close +
        " _Volume: " + this.volume;
}
}
```

C.4 Classe LoadCandleStickForCSVFile

```
package candlesticks;

import handles.CSV_handle;

import java.util.ArrayList;

public class LoadCandleStickForCSVFile
{
    private String Csv_file_path;
    private CSV_handle csv_handle;

    public LoadCandleStickForCSVFile (String Csv_file_path)
    {
        this.Csv_file_path = Csv_file_path;
        csv_handle = new CSV_handle();
    }
}
```

```

    }

    public ArrayList<CandleStick>
    loadCandleStrick ( ArrayList<CandleStick>list )
    {
    csv_handle.readCsvFile ( Csv_file_path , list );
    return list ;
    }
}

```

C.5 Agente experimentalGrafista

```

package Agents;

import java.util.ArrayList;

import javax.swing.JOptionPane;

import patterns.CandleStick.BearishEngulfing;
import patterns.CandleStick.Pattern;
import candlesticks.CandleStick;
import candlesticks.LoadCandleStickForCSVFile;
import jade.core.Agent;
import jade.core.behaviours.Behaviour;
import jade.core.behaviours.OneShotBehaviour;

public class Grafista extends Agent {

    protected void setup ()
    {
        System.out.println ( " Olaheusou um agente grafista "
            + " experimental ....\n" );
        System.out.println ( " Iniciando busca ..... \n"
            + " _____\n" );
        addBehaviour ( new FindPatterns ( this ,
            "C:/ATIVOS/MRVE3SA.csv" ) );
    }

    public class FindPatterns extends OneShotBehaviour
    {

```

```

String path;
publicFindPatterns (Agent a, String path)
{
    super(a);
    this.path=path;
}
@Override
publicvoid action () {

    Pattern testPattern;
    String result = "";
    ArrayList<CandleStick> list
= newArrayList<CandleStick>();
    ArrayList<CandleStick>listResult
= newArrayList<CandleStick>();
    LoadCandleStickForCSVFile test
    = newLoadCandleStickForCSVFile(path);

    test.loadCandleStrick(list);

    testPattern = newBearishEngulfing(list);
    listResult = testPattern.findCandleSticksPatterns();

    JOptionPane.showMessageDialog(null,
        "ListREsult:" + listResult.size());
    JOptionPane.showMessageDialog(null,
        "List:" + list.size());
    if(listResult.size()>0)
System.out.println("Encontrei o padrão nas seguintes datas:...");
    else
System.out.println("Para a atividade não encontrei o padrão buscado");
    for (inti = 0; i<listResult.size(); i++)
    {
        result += listResult.get(i).getDate() + "\n";
    }
    System.out.println(result);
}
}

```

}

APÊNDICE D – Algoritmo Dark Cloud em Python

D.1 Bloco DarkCloud

Created on 02/02/2014

@author: ramon

, , ,

```
def findPatternDarkCloud( list ):
```

```
    # listResult = [ ' ' ]
```

```
    i=2
```

```
    qtd=0
```

```
    C=0
```

```
    O=0
```

```
    H=0
```

```
    L=0
```

```
    C1=0
```

```
    O1=0
```

```
    for candle in list :
```

```
        if (i+1)<len( list ):

```

```
            C=float( list [ i ][ 4 ] )

```

```
            O=float( list [ i ][ 1 ] )

```

```
            H=float( list [ i ][ 2 ] )

```

```
            L=float( list [ i ][ 3 ] )

```

```
            C1=float( list [ i + 1 ][ 4 ] )

```

```
            O1=float( list [ i + 1 ][ 1 ] )

```

```
        i=i+1

```

```
        if ( (C1>O1) and

```

```
            (((C1+O1)/2)>C) and

```

```
            (O>C) and

```

```
            (O>C1) and

```

```

(C>O1) and
(((O-C)/(0.001+H-L))>0.6)
):
qtd=qtd+1
print list[i][0]
print "Frequencia: ",qtd

return

```

D.2 Bloco Bloco FindPatternDarkCloud

```

'''
Created on 01/02/2014

@author: ramon
'''

import csv
import candlesticks
from candlesticks import darkCloud

print __doc__

f=csv.reader(open('/home/ramon/ATIVOS/DTEX3SA.csv'),
              delimiter=',')
conjunto=[]

#Le o arquivo csv

for [date,open,high,low,close,volume,adjclose] in f:
    candlestick=[date,open,high,low,close,volume,adjclose]

    conjunto.append(candlestick)

print f.line_num,'linhas lidas'

#Procura o padrao no historico passado
darkCloud.findPatternDarkCloud(conjunto)

```

APÊNDICE E – Relatório Pesquisa - Ação

001: Coleta de impressões com usuários

E.1 Introdução

Esta coleta de impressões com potenciais usuários teve como principal objetivo verificar o objetivo específico proposto: “Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta”. Vale ressaltar que essa avaliação teve como foco a interface gráfica da ferramenta com os usuários, onde cada usuário recebeu um formulário com questões objetivas e um campo livre para sugestões. A Ferramenta foi apresentada aos potenciais usuários em modo de simulação no período de 1 de março de 2013 a 1 de março de 2014, a versão do JADE utilizado é a 4.3.3 de 10 de dezembro de 2014, a versão do Grais era a 2.4.3 e a versão do java era a 1.7.0_79. A ferramenta foi simulada ainda em um notebook da marca Apple modelo MacBook Pro (Retina, 13-inch, Late 2013) com processador 2.6 GHz Intel Core i5, com 8 GB 1600 MHz DDR3 de Memória RAM e com o Sistema operacional OS X Yosemite versão 10.10.3

E.2 Participantes

A avaliação ocorreu em maio de 2014 e participaram da avaliação 6 participante onde: 1 classifica-se no perfil Corajoso; 1 classifica-se no perfil Moderado e 4 classificam-se no perfil Conservador. Foi entregue a cada participante um formulário não identificado com 7 questões objetivas e uma questão aberta.

E.3 Impressões Coletadas

Quando questionados sobre as cores adotadas na ferramenta, 5 pessoas consideraram boas e 1 pessoa considerou ótimas. Quando questionados sobre a facilidade de compreensão da ferramenta, 3 pessoas consideraram como bom, 1 pessoa considerou razoável, 1 pessoa considerou como ruim e 1 se absteve. Quando solicitados observações gerais para ferramenta destacaram-se: (i) feedback indicando o carregamento e o processo da ferramenta; (ii) atualizar página principal automaticamente; (iii) dispor de um canal de dúvidas; (iv) não adotar estratégias baseadas em Médias Móveis; (v) utilizar javascript na interface; (vi) Executar a ferramenta em duas máquinas; (vii) Informações sobre estratégias e uso da ferramenta.

As sugestões apresentadas pelos usuários geram demanda por melhorias na interface gráfica, como esperado, e uma demanda na melhoria nas estratégias adotadas, esta demanda coincidiu com os resultados obtidos no Relatório 02, onde verificou-se que o uso de estratégias baseadas em médias móveis para o perfil Conservador se mostrou ineficiente. Para atender a demanda de melhorias na interface gráfica foi necessário realizar pequenas capacitações em tecnologias front-end, tais como JavaScript e jQuery. Para atender a demanda de melhorias nas estratégias, foi necessário criar novas estratégias baseadas em outros indicadores financeiros e simular novamente a estratégia.

E.4 Ações Tomadas

Foi criado um mecanismo de onde o usuário tem a possibilidade de enviar dúvidas, sugestões, reclamações e elogios. Foi criada uma página com informações sobre a ferramenta bem como suas estratégias. Foi adicionado no projeto grails o plugin JQuery UI v.1.10.4 e jQuery v.1.11.1.

E.5 Conclusão

Foram identificadas nesta coleta de impressões, demandas por melhorias onde para atender foi necessário buscar tecnologias front-end que auxiliassem a implementação. Foi implementado ainda um mecanismo por onde o usuário tem a possibilidade de enviar dúvidas, sugestões, reclamações e elogios. Foi criado ainda uma página com informações sobre as estratégias adotadas pela ferramenta. Por fim, pode-se concluir que esta coleta de impressões obteve sucesso.

APÊNDICE F – Relatório Pesquisa - Ação

001: Coleta de Impressões através de simulação por perfil de usuário

F.1 Introdução

Esta coleta de impressões com potenciais usuários teve como principal objetivo verificar o objetivo específico proposto: “Abstrair a complexidade de cálculos financeiros comumente utilizados na Análise Técnica. Assim, essa complexidade não é sentida pelo usuário, deixando a mesma a cargo da ferramenta”. Vale ressaltar que essa avaliação teve como foco as estratégias adotadas na ferramenta.

As estratégias foram simuladas no período no período de 1 de janeiro de 2012 a 1 de março de 2015, a versão do JADE utilizado é a 4.3.3 de 10 de dezembro de 2014, a versão do Grais era a 2.4.3 e a versão do java era a 1.7.0_79. A ferramenta foi simulada ainda em um notebook da marca Apple modelo MacBook Pro (Retina, 13-inch, Late 2013) com processador 2.6 GHz Intel Core i5, com 8 GB 1600 MHz DDR3 de Memória RAM e com o Sistema operacional OS X Yosemite versão 10.10.3.

F.2 Impressões coletadas

Todas estratégias de todos perfis foram simuladas de maneira conjunta de acordo com o perfil relacionado, as tabelas apresentadas abaixo contém os dados detalhados das simulações.

F.3 Dados coletados

F.3.1 Dados Simulação Perfil Conservador

Estratégias Perfil Conservador e Resultados

Agente	estratégia	Percentual de Ganho
A1	Média Móvel Simples 13/21	-29,06%
A2	Media Móvel Simples 21/34	-31,74%

Continuação na próxima página

Tabela 17 – Continuação da página anterior

Agente	estratégia	Percentual de Ganho
A3	Media Móvel Exponencial 21/34	34,96%
A4	Media Móvel Simples 13/21	4,93%
A5	Media Móvel Simples 21/34.	-60,41%
A6	Media Móvel Exponencial 21/34.	8,26%
A7	Media Móvel Simples 13/21.	59,91%
	TOTAL GERAL	7,8%

Agente A1: Ação GEPA3.SA

DATA	Ordem	R\$	Ganho (%)
1/5/2013	Buy	64.02	
3/0/2014	Buy	62.04	
1/2/2014	Sell	47.51	-26.00%
1/4/2014	Buy	48	
		% Acumulado	-26,00%

Agente A1: Ação SHOW3.SA

DATA	Ordem	R\$	Ganho (%)
5/1/2012	Buy	13.2	
4/4/2012	Sell	15.5	17.42%
5/6/2012	Buy	17.37	
2/7/2012	Sell	17.3	-0.40%
5/7/2012	Buy	17.39	
3/8/2012	Sell	14.1	-18.92%
1/0/2013	Buy	7.62	
5/1/2013	Sell	7.68	0.79%
3/2/2013	Buy	7.9	

Continuação na próxima página

Tabela 19 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
4/4/2013	Sell	8.44	6.84%
2/4/2013	Buy	9.19	
4/5/2013	Sell	6.86	-25.35%
2/7/2013	Buy	8.59	
3/8/2013	Sell	7.7	-10.36%
4/11/2013	Buy	5.35	
4/1/2014	Sell	4.8	-10.28%
4/3/2014	Buy	4.93	
2/4/2014	Sell	5	1.42%
4/4/2014	Buy	4.88	
2/5/2014	Sell	5.2	6.56%
2/6/2014	Buy	5.33	
4/7/2014	Sell	5.1	-4.32%
4/11/2014	Buy	2.88	
5/0/2015	Sell	2.98	3.47%
		% Acumulado	-26,00%

Agente A1: Ação TELB4.SA

DATA	Ordem	R\$	Ganho (%)
5/1/2012	Buy	14.79	
4/3/2012	Sell	13.69	-7.44%
1/9/2012	Buy	7.37	
3/11/2012	Sell	6.61	-10.31%
4/11/2012	Buy	8.04	
1/0/2013	Sell	7.09	-11.82%
5/9/2013	Buy	2.87	
4/9/2013	Sell	2.7	-5.92%
5/10/2013	Buy	2.82	
4/10/2013	Sell	2.84	0.71%
5/3/2014	Buy	2.52	

Continuação na próxima página

Tabela 20 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
5/4/2014	Sell	2.37	-5.95%
5/5/2014	Buy	2.49	
4/6/2014	Sell	2.33	-6.43%
2/8/2014	Buy	2.26	
4/9/2014	Sell	1.87	-17.26%
5/2/2015	Buy	1.13	
		% Acumulado	27,3%

Agente A2: Ação TERI3.SA

DATA	Ordem	R\$	Ganho (%)
3/4/2012	Sell	0.22	
3/8/2012	Buy	0.16	
3/10/2012	Sell	0.13	-18.75%
3/6/2013	Buy	0.12	
4/10/2013	Sell	0.12	0.00%
3/1/2014	Buy	0.11	
5/1/2014	Sell	0.11	0.00%
3/5/2014	Buy	0.1	
2/6/2014	Sell	0.09	-10.00%
4/6/2014	Buy	0.1	
4/6/2014	Sell	0.1	0.00%
3/8/2014	Buy	0.09	
5/8/2014	Sell	0.09	0.00%
		% Acumulado	-28,75%

Agente A2: Ação RPMG3.SA

DATA	Ordem	R\$	Ganho (%)
1/4/2012	Buy	0.11	
2/4/2012	Sell	0.1	-9.09%
4/3/2013	Buy	0.03	
2/4/2013	Sell	0.03	0.00%
2/5/2013	Buy	0.03	
2/10/2013	Sell	0.03	0.00%
3/11/2014	Buy	0.33	
2/2/2015	Sell	0.25	-24.24%
		% Acumulado	-33,33%

Agente A2: Ação SCLO4.SA

DATA	Ordem	R\$	Ganho (%)
5/1/2012	Sell	3.79	
1/1/2012	Buy	3.89	
3/1/2012	Sell	3.7	-4.88%
4/3/2012	Buy	3.85	
2/3/2012	Sell	3.85	0.00%
2/4/2012	Buy	3.85	
		% Acumulado	-4,88%

Agente A2: Ação TXRX4.SA

DATA	Ordem	R\$	Ganho (%)
4/2/2012	Buy	0.44	
1/9/2012	Sell	0.27	-38.64%
5/6/2014	Buy	0.28	
4/0/2015	Sell	0.22	-21.43%

Continuação na próxima página

Tabela 24 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
4/2/2015	Buy	0.28	
		% Acumulado	-60,07%

Agente A3: Ação MGEL4.SA

DATA	Ordem	R\$	Ganho (%)
2/8/2013	Sell	0.22	
4/9/2013	Buy	0.17	
5/2/2014	Sell	0.12	-29.41%
2/2/2014	Buy	0.11	
1/2/2014	Sell	0.12	9.09%
4/3/2014	Buy	0.11	
5/5/2014	Sell	0.2	81.82%
5/9/2014	Buy	0.15	
3/1/2015	Sell	0.14	-6.67%
		% Acumulado	54,83%

Agente A3: Ação SPRI3.SA

DATA	Ordem	R\$	Ganho (%)
3/10/2012	Sell	0.44	
3/2/2013	Buy	0.49	
1/8/2013	Sell	0.42	-14.29%
2/9/2013	Buy	0.36	
1/0/2014	Sell	0.39	8.33%
3/2/2014	Buy	0.37	
		% Acumulado	-5,96%

Agente A3: Ação TCNO3.SA

DATA	Ordem	R\$	Ganho (%)
1/6/2013	Sell	0.16	
5/10/2013	Buy	0.08	
2/0/2014	Sell	0.09	12.50%
4/2/2014	Buy	0.08	
4/3/2014	Sell	0.13	62.50%
4/5/2014	Buy	0.1	
1/5/2014	Sell	0.11	10.00%
4/9/2014	Buy	0.22	
2/2/2015	Sell	0.14	-36.36%
		% Acumulado	48,64%

Agente A4: Ação BPHA3.SA

DATA	Ordem	R\$	Ganho (%)
2/4/2012	Sell	9.32	
3/6/2012	Buy	10.93	
2/9/2012	Sell	12.35	12.99%
1/10/2012	Buy	12.69	
1/2/2013	Sell	13.9	9.54%
2/9/2013	Buy	8.55	
1/10/2013	Sell	7.45	-12.87%
1/5/2014	Buy	3.73	
1/6/2014	Sell	3.3	-11.53%
2/7/2014	Buy	3.89	
1/9/2014	Sell	3.77	-3.08%
		% Acumulado	4,95%

Agente A4: Ação CBEE3.SA

DATA	Ordem	R\$	Ganho (%)
2/1/2012	Buy	1.8	
4/3/2012	Sell	1.31	-27.22%
1/4/2012	Buy	1.7	
1/5/2012	Sell	1.58	-7.06%
1/7/2012	Buy	1.46	
3/9/2012	Sell	1.3	-10.96%
5/4/2013	Buy	1.75	
4/6/2013	Sell	1.16	-33.71%
4/7/2013	Buy	1.33	
3/9/2013	Sell	1.29	-3.01%
2/11/2013	Buy	1.29	
3/0/2014	Sell	1.2	-6.98%
4/3/2014	Buy	1.14	
2/4/2014	Sell	1.2	5.26%
4/4/2014	Buy	1.24	
5/6/2014	Sell	1.25	0.81%
5/7/2014	Buy	1.23	
1/7/2014	Sell	1.2	-2.44%
3/8/2014	Buy	1.16	
4/8/2014	Sell	1.15	-0.86%
1/11/2014	Buy	1.05	
1/0/2015	Sell	1.09	3.81%
3/0/2015	Buy	1	
1/1/2015	Sell	1.09	9.00%
5/1/2015	Buy	0.95	
		% Acumulado	-73,36%

Agente A4: Ação HETA4.SA

DATA	Ordem	R\$	Ganho (%)
5/2/2012	Sell	0.37	

Continuação na próxima página

Tabela 30 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
1/7/2012	Buy	0.25	
4/8/2012	Sell	0.23	-8.00%
2/0/2013	Buy	0.16	
4/0/2013	Sell	0.17	6.25%
3/4/2013	Buy	0.14	
1/6/2013	Sell	0.28	100.00%
4/8/2013	Buy	0.32	
1/9/2013	Sell	0.27	-15.63%
4/11/2013	Buy	0.31	
3/1/2014	Sell	0.31	0.00%
1/6/2014	Buy	0.33	
2/9/2014	Sell	0.4	21.21%
3/2/2015	Buy	0.34	
		% Acumulado	103,83%

Agente A4: Ação MLFT4.SA

DATA	Ordem	R\$	Ganho (%)
3/5/2012	Sell	1.7	
1/7/2012	Buy	1.57	
1/8/2012	Sell	1.55	-1.27%
2/9/2012	Buy	1.69	
4/11/2012	Sell	1.57	-7.10%
1/0/2013	Buy	1.7	
1/3/2013	Sell	1.72	1.18%
5/9/2013	Buy	1.58	
2/10/2013	Sell	1.5	-5.06%
2/11/2013	Buy	1.6	
4/1/2014	Sell	1.6	0.00%
3/7/2014	Buy	1.4	
3/7/2014	Sell	1.34	-4.29%

Continuação na próxima página

Tabela 31 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
3/9/2014	Buy	1.36	
2/10/2014	Sell	1.22	-10.29%
		% Acumulado	-28,83%

Agente A4: Ação VVAR3.SA

DATA	Ordem	R\$	Ganho (%)
2/3/2012	Sell	20	
4/7/2012	Buy	19	
1/8/2012	Sell	18	-5.26%
1/11/2012	Buy	18	
1/0/2013	Sell	18	0.00%
2/1/2013	Buy	18.02	
2/6/2013	Sell	23	27.64%
2/7/2013	Buy	24	
5/10/2013	Sell	8.97	-62.63%
5/3/2014	Buy	7.1	
1/6/2014	Sell	7.55	6.34%
4/8/2014	Buy	8.09	
5/9/2014	Sell	7	-13.47%
4/10/2014	Buy	7.5	
4/11/2014	Sell	5.13	-31.60%
5/0/2015	Buy	7.38	
		% Acumulado	-78,98%

Agente A5: Ação JHSF3.SA

DATA	Ordem	R\$	Ganho (%)
3/4/2012	Sell	5.96	
2/7/2012	Buy	6.75	
1/2/2013	Sell	8	18.52%
4/8/2013	Buy	6.72	
4/9/2013	Sell	5.58	-16.96%
4/1/2014	Buy	4.1	
1/2/2014	Sell	4	-2.44%
5/6/2014	Buy	4	
5/8/2014	Sell	3.84	-4.00%
		% Acumulado	-4,88%

Agente A5: Ação MMXM3.SA

DATA	Ordem	R\$	Ganho (%)
5/4/2012	Sell	8.18	
3/0/2013	Buy	3.95	
5/1/2013	Sell	3.3	-16.46%
3/7/2013	Buy	1.99	
4/9/2013	Sell	1.07	-46.23%
5/0/2014	Buy	3.7	
2/3/2014	Sell	2.54	-31.35%
2/11/2014	Buy	0.59	
5/2/2015	Sell	0.66	11.86%
		% Acumulado	-82,18%

Agente A5: Ação OGXP3.SA

DATA	Ordem	R\$	Ganho (%)
4/4/2012	Sell	13.75	
1/8/2012	Buy	6.63	
3/9/2012	Sell	4.63	-30.17%
2/0/2014	Buy	0.31	
5/2/2014	Sell	0.25	-19.35%
2/6/2014	Buy	0.19	
4/7/2014	Sell	0.18	-5.26%
		% Acumulado	-54,78%

Agente A5: Ação VIVR3.SA

DATA	Ordem	R\$	Ganho (%)
5/1/2012	Buy	2.14	
3/4/2012	Sell	1.93	-9.81%
4/1/2013	Buy	1.04	
5/3/2013	Sell	0.63	-39.42%
5/9/2013	Buy	0.32	
4/0/2014	Sell	0.25	-21.88%
1/7/2014	Buy	0.21	
3/9/2014	Sell	0.12	-42.86%
		% Acumulado	-113,97%

Agente A6: Ação CRIV4.SA

DATA	Ordem	R\$	Ganho (%)
5/2/2012	Sell	4.32	
3/5/2012	Buy	3.99	
5/7/2012	Sell	4.05	1.50%

Continuação na próxima página

Tabela 37 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
2/5/2013	Buy	5.19	
		% Acumulado	1,5%

Agente A6: Ação MWET4.SA

DATA	Ordem	R\$	Ganho (%)
4/8/2012	Sell	1.18	
4/10/2012	Buy	1.12	
2/3/2013	Sell	0.94	-16.07%
4/3/2013	Buy	0.78	
1/3/2013	Sell	1.1	41.03%
2/6/2013	Buy	0.91	
5/7/2013	Sell	1.09	19.78%
3/8/2013	Buy	0.91	
2/8/2013	Sell	0.97	6.59%
4/1/2014	Buy	1.35	
2/1/2014	Sell	1.6	18.52%
1/1/2014	Buy	1.46	
5/1/2014	Sell	1.56	6.85%
2/2/2014	Buy	1.38	
5/2/2014	Sell	1.63	18.12%
5/3/2014	Buy	1.47	
		% Acumulado	94,82%

Agente A6: Ação RJCP3.SA

DATA	Ordem	R\$	Ganho (%)
3/9/2013	Sell	0.02	

Continuação na próxima página

Tabela 39 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
1/11/2013	Buy	0.01	
		% Acumulado	0,0%

Agente A6: Ação SNSY5.SA

DATA	Ordem	R\$	Ganho (%)
1/1/2012	Sell	4.06	
2/4/2012	Buy	3.66	
5/7/2012	Sell	3.26	-10.93%
4/9/2012	Buy	2.81	
2/9/2013	Sell	1.26	-55.16%
4/10/2013	Buy	0.95	
4/3/2014	Sell	0.87	-8.42%
3/4/2014	Buy	0.8	
5/7/2014	Sell	0.89	11.25%
2/7/2014	Buy	0.81	
		% Acumulado	-63,26%

Agente A7: Ação CBMA4.SA

DATA	Ordem	R\$	Ganho (%)
5/2/2012	Sell	0.08	
2/7/2012	Buy	0.07	
5/8/2012	Sell	0.05	-28.57%
1/9/2012	Buy	0.07	
5/11/2012	Sell	0.07	0.00%
3/1/2013	Buy	0.06	
5/2/2013	Sell	0.05	-16.67%

Continuação na próxima página

Tabela 41 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
1/4/2013	Buy	0.05	
3/5/2013	Sell	0.05	0.00%
3/6/2013	Buy	0.05	
1/8/2013	Sell	0.04	-20.00%
3/10/2013	Buy	0.04	
4/0/2014	Sell	0.05	25.00%
1/1/2014	Buy	0.04	
2/2/2014	Sell	0.04	0.00%
5/6/2014	Buy	0.05	
4/9/2014	Sell	0.05	0.00%
		% Acumulado	-40,24%

Agente A7: Ação DTCY3.SA

DATA	Ordem	R\$	Ganho (%)
5/2/2012	Sell	0.91	
4/4/2012	Buy	0.9	
3/4/2012	Sell	0.8	-11.11%
1/8/2012	Buy	0.66	
2/9/2012	Sell	0.57	-13.64%
2/1/2013	Buy	0.66	
2/3/2013	Sell	0.59	-10.61%
4/3/2013	Buy	0.59	
3/4/2013	Sell	0.51	-13.56%
1/7/2013	Buy	0.43	
1/8/2013	Sell	0.4	-6.98%
2/0/2014	Buy	0.4	
3/2/2014	Sell	0.31	-22.50%
3/4/2014	Buy	0.4	
4/7/2014	Sell	0.45	12.50%
4/8/2014	Buy	0.55	

Continuação na próxima página

Tabela 42 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
3/8/2014	Sell	0.51	-7.27%
2/11/2014	Buy	0.39	
1/11/2014	Sell	0.38	-2.56%
		% Acumulado	-75,73%

Agente A7: Ação GPCP3.SA

DATA	Ordem	R\$	Ganho (%)
1/2/2012	Sell	0.43	
3/6/2012	Buy	0.37	
3/7/2012	Sell	0.34	-8.11%
5/0/2013	Buy	0.16	
4/1/2013	Sell	0.16	0.00%
2/1/2013	Buy	0.15	
1/1/2013	Sell	0.15	0.00%
3/1/2013	Buy	0.15	
5/2/2013	Sell	0.15	0.00%
4/5/2013	Buy	0.09	
3/10/2013	Sell	0.41	355.56%
1/11/2013	Buy	0.45	
2/0/2014	Sell	0.5	11.11%
5/1/2014	Buy	0.49	
4/2/2014	Sell	0.47	-4.08%
4/2/2014	Buy	0.54	
		% Acumulado	354,48%

Agente A7: Ação PRML3.SA

DATA	Ordem	R\$	Ganho (%)
1/3/2012	Sell	3.39	
3/6/2012	Buy	2.41	
5/8/2012	Sell	2.88	19.50%
1/11/2012	Buy	2.4	
1/1/2013	Sell	2.02	-15.83%
1/2/2013	Buy	2.02	
4/3/2013	Sell	1.87	-7.43%
5/7/2013	Buy	1.6	
1/9/2013	Sell	1.25	-21.88%
2/0/2014	Buy	1.09	
3/0/2014	Sell	0.96	-11.93%
1/3/2014	Buy	1.22	
3/5/2014	Sell	1.1	-9.84%
2/7/2014	Buy	1.1	
2/8/2014	Sell	1.02	-7.27%
		% Acumulado	-54,68%

Agente A7: Ação RCSL4.SA

DATA	Ordem	R\$	Ganho (%)
5/2/2012	Sell	0.23	
2/8/2012	Buy	0.1	
4/8/2012	Sell	0.08	-20.00%
4/7/2013	Buy	0.05	
4/8/2013	Sell	0.04	-20.00%
1/8/2013	Buy	0.05	
5/11/2013	Sell	0.08	60.00%
5/1/2014	Buy	0.08	
4/2/2014	Sell	0.06	-25.00%
3/4/2014	Buy	0.07	
3/5/2014	Sell	0.06	-14.29%

Continuação na próxima página

Tabela 45 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
2/7/2014	Buy	0.05	
5/8/2014	Sell	0.05	0.00%
4/11/2014	Buy	0.04	
1/11/2014	Sell	0.03	-25.00%
4/2/2015	Buy	0.04	
		% Acumulado	-44,29%

F.3.2 Dados Simulação Perfil Corajoso

Estratégias Perfil Corajoso e Resultados

Agente	estratégia	Percentual de Ganho
A1	Bearish - Bullish	-34,71%
A2	Dark Cloud - Bullish Engulf	-6,45%
	TOTAL GERAL	-18,08%

Agente A1: Ação BMKS3.SA

DATA	Ordem	R\$	Ganho (%)
05/02/12	Sell	329.9	
06/19/12	Buy	305.95	
06/24/12	Sell	295	-3.58%
08/14/12	Buy	305	
08/28/12	Sell	298	-2.30%
09/05/12	Sell	290	
03/19/13	Sell	362	
04/01/13	Buy	375	
05/02/13	Sell	351	-6.40%

Continuação na próxima página

Tabela 47 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
09/02/13	Buy	330	
09/03/13	Sell	320.06	-3.012%
09/23/13	Buy	340	
11/20/13	Sell	331	-2.647%
01/14/14	Buy	340	
03/07/14	Sell	320.5	-5.735%
03/14/14	Buy	350	
03/25/14	Sell	337	-3.714%
03/26/14	Buy	350	
03/27/14	Sell	330	-5.714%
04/04/14	Buy	350	
04/18/14	Sell	344	
04/24/14	Buy	349	
04/28/14	Sell	330	-5.444%
07/24/14	Buy	310	
08/04/14	Sell	299	-3.548%
08/13/14	Buy	305	
09/10/14	Sell	300	-1.639%
09/23/14	Buy	320	
10/02/14	Sell	285	-10.938%
11/28/14	Buy	298	
12/10/14	Sell	270	-9.396%
01/16/14	Buy	288.98	
01/29/14	Sell	260.25	-9.942%
02/19/15	Buy	275	
		% Acumulado	-54,68%

Agente A1: Ação DAGB11.SA

DATA	Ordem	R\$	Ganho (%)
02/02/12	Sell	204	

Continuação na próxima página

Tabela 48 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
03/22/12	Buy	232	
04/19/12	Buy	240	
05/24/12	Sell	237	-1.25%
06/15/12	Buy	242	
08/09/12	Sell	243	0.413%
08/14/12	Buy	251	
09/28/12	Sell	242.5	-3.386%
12/14/12	Sell	267	
12/17/12	Buy	274	
01/01/12	Sell	272	-0.730%
01/03/12	Sell	268	
02/12/13	Sell	281.78	
04/23/13	Buy	269.5	
06/11/13	Buy	288	
10/10/13	Sell	322	11.806%
01/20/14	Sell	400.5	
02/20/14	Buy	395.99	
02/24/14	Sell	387	-2.270%
03/03/14	Buy	396	
06/02/14	Buy	375	
		% Acumulado	4,58%

Agente A2: Ação KLBN4.SA

DATA	Ordem	R\$	Ganho (%)
03/13/12	Buy	9.63	
09/21/12	Sell	10.07	4.569%
09/25/12	Sell	10.01	
09/27/12	Sell	10.3	
11/16/12	Buy	11.85	
11/21/12	Buy	12.14	

Continuação na próxima página

Tabela 49 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
11/23/12	Buy	12	
11/26/12	Buy	11.8	
11/27/12	Buy	11.7	
11/28/12	Buy	12	
11/29/12	Buy	12	
11/30/12	Buy	12.04	
04/05/13	Buy	14.29	
04/08/13	Buy	14.42	
04/09/13	Buy	14.41	
04/10/13	Buy	14.53	
04/12/13	Buy	13.76	
04/15/13	Buy	13.53	
04/16/13	Buy	13.76	
04/17/13	Buy	13.34	
04/18/13	Buy	13.43	
		% Acumulado	4,00%

Agente A2: Ação USIM3.SA

DATA	Ordem	R\$	Ganho (%)
09/19/12	Buy	13.49	
09/20/12	Buy	13.29	
09/24/12	Buy	13.2	
09/25/12	Buy	11.7	
09/26/12	Buy	12.22	
09/28/12	Buy	11.62	
10/01/12	Buy	11.74	
11/28/12	Buy	12.66	
11/29/12	Buy	13.6	
11/30/12	Buy	13.65	
11/03/12	Buy	14.06	

Continuação na próxima página

Tabela 50 – Continuação da página anterior

DATA	Ordem	R\$	Ganho (%)
12/04/12	Buy	13.85	
11/05/12	Buy	13.16	
12/06/12	Buy	12.92	
12/07/12	Buy	13.6	
12/10/12	Buy	13.67	
12/11/12	Buy	13.52	
02/06/12	Sell	11.2 -16.976%	
02/08/13	Sell	11.33	
02/11/13	Sell	11.33	
07/31/13	Sell	8.77	
08/01/13	Sell	8.76	
08/02/13	Sell	8.6	
08/05/13	Sell	8.77	
08/06/13	Sell	8.5	
08/07/13	Sell	8.67	
08/08/13	Sell	9.09	
08/09/13	Sell	9.58	
08/12/13	Sell	9.58	
01/03/14	Buy	12.35	
01/06/14	Buy	12.49	
01/07/14	Buy	12.54	
01/08/14	Buy	12.39	
01/09/14	Buy	12.21	
		% Acumulado	-16,9%

F.4 Ações tomadas

As estratégias financeiras do perfil Corajoso, associadas aos agentes especialistas foram substituídas por estratégias baseadas em Médias Moveis Exponenciais com uso de *Stop Loss*. Para esta versão da Ferramenta de Software, não serão utilizadas as estratégias baseadas em padrões de Candlestick. Verificou-se, nesse caso, que para desenvolver um código capaz de identificar variações de um padrão de Candlestick, bem como vali-

dar os sinais, se falsos ou não, seria mais adequado usar uma abordagem de inteligência artificial orientada a mecanismos de aprendizagem. Tal mecanismo não é escopo deste trabalho, entretanto, trata-se de uma sugestão de melhoria para a ferramenta. Essa melhoria poderá usufruir da abordagem evolutiva adotada no desenvolvimento dessa ferramenta, ampliando ainda mais a capacidade de atuação dos agentes.

APÊNDICE G – Documento de Arquitetura

G.1 Introdução

Neste documento é detalhado as principais partes da arquitetura implementada para a Ferramenta de Software desenvolvida. A arquitetura é composta por padrões de projetos orientados a objetos e por agentes comportamentais de Software. O Padrão Arquitetural adotado é o Model-View-Controller adaptado ao uso do paradigma Orientado a Agentes Comportamentais. Este documento contém ainda, uma proposta de refatoração arquitetural no tocante aos Agentes implementados, tais sugestões foram obtidas através da experiência do desenvolvedor com o paradigma e como ele é interpretado pela plataforma JADE.

G.2 Objetivos

Este Documento de Arquitetura da Ferramenta de Software provê uma visão geral da arquitetura, usando um conjunto de visões para tratar aspectos específicos do paradigma abordado.

Este documento serve como um subsidio para posteriores manutenções evolutivas ou corretivas da Ferramenta de Software.

G.3 Arquiterura desenhada

G.3.1 Visão Geral

Esta Ferramenta de Software foi desenvolvida com o propósito de auxiliar um cidadão brasileiro a comprar boas ações e vender-las de maneira que obtenha o maior lucro possível, dentro do perfil de usuário escolhido. A Ferramenta proposta está contextualizada na Bolsa de Valores de São Paulo e esta faz uso de estratégias financeiras baseadas em análise técnica de Ações da Bolsa. Ao identificar alguma Ação para compra, ela sugere ao usuário sua compra e posteriormente sua venda.

O Objetivo Geral da Arquitetura é *“Prover uma Ferramenta de Estratégia Financeira apoiada por Sistemas Multiagentes Comportamentais”*. E os objetivos Específicos atendidos pela Arquitetura da Ferramenta são : (i) Abstrair a complexidade de Cálculos financeiros comumente utilizados na Análise Técnica; e (ii) Explorar o Paradigma de Sistemas Multiagentes Comportamentais baseados em tempo, no contexto financeiro. Para

tal, foram criadas as seguintes Estórias de Usuário, estas nortearam o desenvolvimento:

1. Como engenheiro, quero um agente capaz de montar e gerir uma carteira de ações de acordo com o perfil de investidos escolhido pelo usuário e valor investido.
2. Como engenheiro, quero um agente capaz de tomar estratégias financeiras de acordo com o perfil de investidor escolhido pelo usuário e valor investido.

Para atender estas estórias de usuário, assim como as demais estórias do Backlog da Ferramenta de Software desenvolvida, foram criadas Camadas de Software. A primeira camada, camada de Visão, abriga os código relacionados a tecnologia Web, Grails. Esta responsável por ser a principal interface com o usuário final. As segunda e terceira camadas, Camada de Controle e Modelo, abrigam os código relacionados ao servidor, escrito em Linguagem JAVA+JADE.

A Camada de Controle contém o pacote core. No pacote core estão concentrados os códigos que implementam os agentes comportamentais propostos e suas classes de suporte, concentra-se neste pacote de maneira adicional, uma estrutura proposta para futura refatoração arquitetural da ferramenta, no tocante aos comportamentos dos agente. Os tópicos seguintes apresentam mais detalhes dos agentes implementados contidos neste pacote.

A Camada de Modelo contém o pacote suport, este dividido em três subpacotes - financeiro, util e statistical. De maneira sucinta, as classes contidas no subpacote financeiro são responsáveis por prover o mecanismo de estratégias financeiras a serem utilizadas pelos agentes comportamentais da Ferramenta de Software. As classes contidas no subpacote statistical são responsáveis por prover mecanismos de calculos estatísticos utilizados por agentes do tipo Caçador (Hunter) para selecionar Ações compatíveis com os perfis abordados pela Ferramenta, elas são utilizadas ainda pelos agentes do tipo Gestor (Manager) para realizar o controle de risco envolvido em uma carteira de Ações. As classes contidas no pacote util, são responsáveis por prover o mecanismo de persistência de dados no MongoDB. Vale ressaltar que a comunicação entre as camadas de Visão e Controle é feita via banco de dados, como ilustrado na figura 2.

G.3.2 Pacote Core

O pacote core contém as classes que implementam os Agentes comportamentais utilizados na Ferramenta de Software desenvolvida. Ele contém o subpacote agent e este por sua vez contém os subpacotes behaviours, util e suport. O subpacote util contém classes responsáveis por realizar testes funcionais de comportamentos de agentes, como apresentado na figura 3. As Classes AgentA e AgentB tem o propósito de realizar testes de comunicação e comportamentos, esta estratégia foi necessária uma vez que a plataforma



Usuário

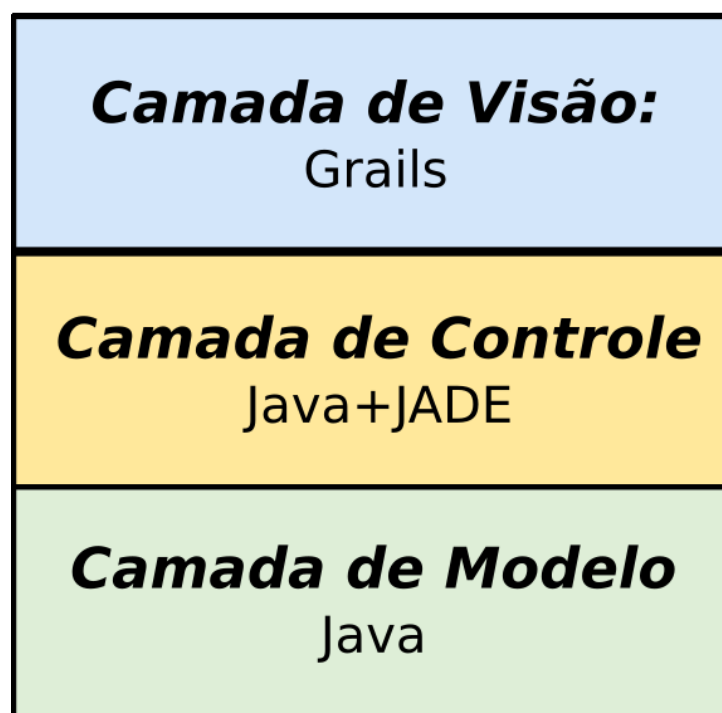


Figura 01:Arquitetura visão Geral

JADE até a data de publicação deste documento carecia uma estrutura adequada para testes de comportamentos. As Classes StockAgent, SimulationBehaviour, StocksInMemory e Simulation Setup compõe uma estrutura pensada pelo desenvolvedor para simular a Ferramenta quando não houver conexão com a internet.

O pacote behaviours contém comportamentos isolados a serem agregados a agentes que necessitarem deles. Durante o desenvolvimento da Ferramenta proposta e após seguir as práticas adotadas pelas literaturas relacionadas à implementação de agentes com o JADE, o desenvolvedor elaborou uma arquitetura baseada em interfaces de maneira a reduzir o acoplamento de classes que implementam agentes comportamentais, bem como facilitar futuras manutenções evolutivas e corretivas. Esta arquitetura contém uma interface denominada ProcedureBehaviour e uma classe concreta denominada Com-

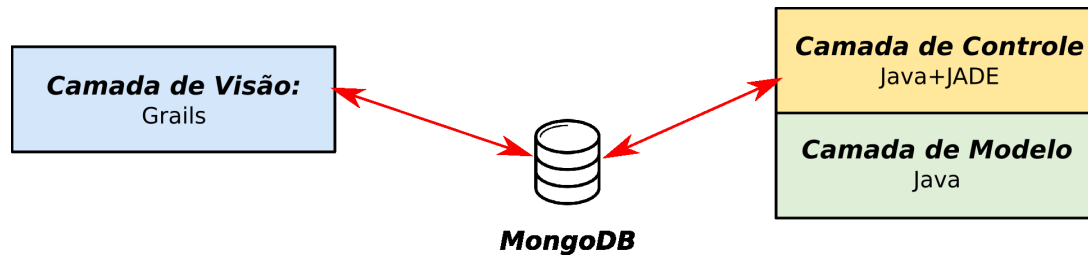


Figura 02: Comunicação entre camadas

municationBehaviour, e é uma sugestão de manutenção evolutiva para a Ferramenta de Software, figura 4.

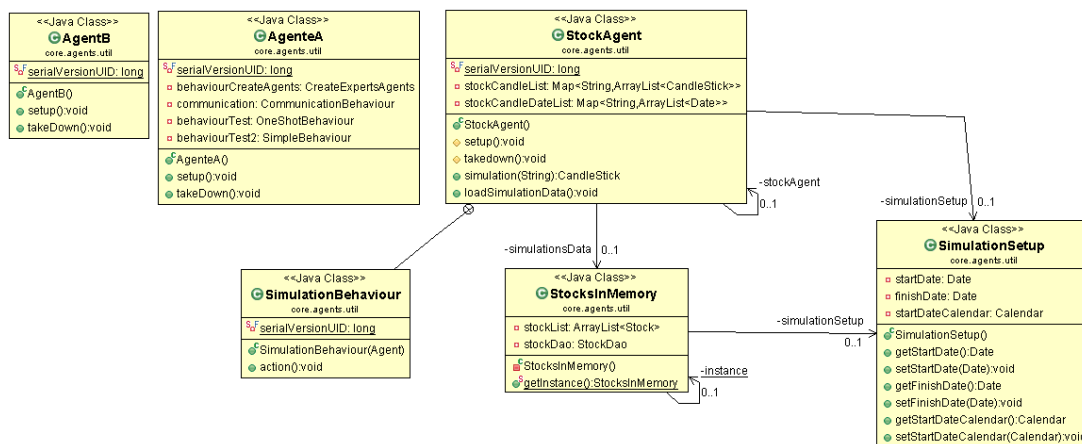


Figura 03: Classes de Apoio

A Classe CommunicationBehaviour implementa o comportamento em comum entre os agentes comportamentais que é receber mensagens de outros agentes e responder estas mensagens através da execução de um comportamento. Esta classe dispõe de um mecanismo de adição de comportamentos, na qual para que essa adição seja feita é necessário informar a classe, os atributos ConversationID e uma classe concreta ProcedureBehaviour. Um classe agente comportamental pode utilizar uma classe ProcedureBehaviour sem a necessidade de utilizar a classe CommunicationBehaviour. Esta classe agente pode utilizar da agregação para aplicar o seu conjunto de comportamentos, assim reduzindo o acoplamento da classe e aumentando sua coesão.

G.3.2.1 Agentes implementados

Conforme proposto no capítulo 4, Foram implementados os 3 tipos de agentes de Software planejados: (i) Agente Gestor; (ii) Agente Especialista; (iii) Agente Caçador. No entanto, verificou-se a necessidade em criar um quarto tipo de Agente de Software, o Agente Criador. Este responsável por auxiliar na comunicação entre usuários web com seus respectivos Agentes de Software.

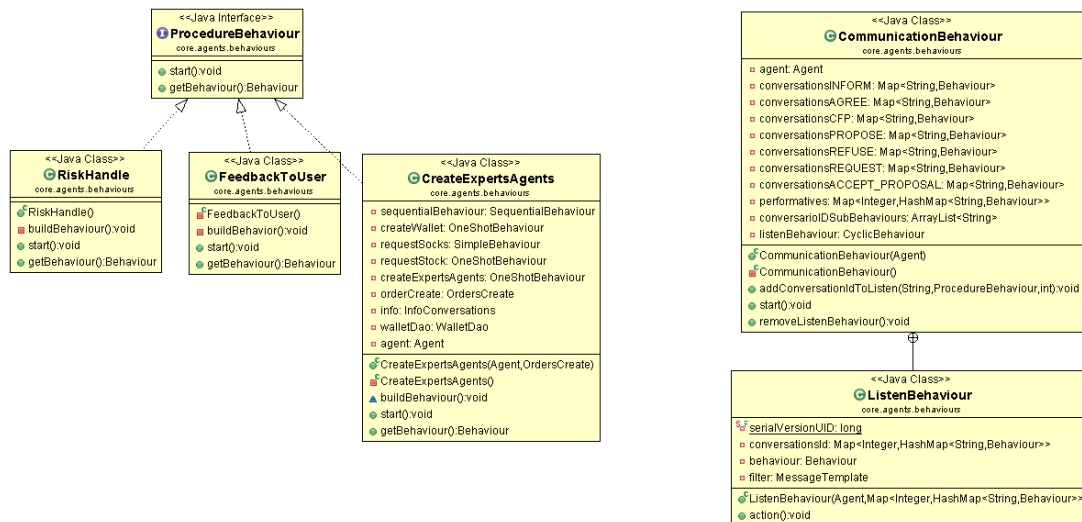


Figura 04: Arquitetura sugerida para refatoração arquitetural

G.3.2.1.1 Agente Gestor (Manager)

O Agente do tipo Gestor é o responsável por administrar a carteira de ações dos usuários, ações que por sua vez são acompanhadas por Agentes Especialistas. Ele autoriza uma compra ou venda de ações analisadas pelos Agentes Especialistas, vale ressaltar que essa autorização passa pelo usuário vinculado ao grupo de Agentes. O Agente Gestor, de maneira autônoma, faz o monitoramento de risco envolvido na carteira de ações administrada. Bem como, executa o processo de montagem de carteira de maneira alinhada com o perfil escolhido pelo usuário. De acordo com as responsabilidades descritas no capítulo 4, foram implementadas as seguintes responsabilidades:

- Criar e excluir um ou mais Agentes Especialistas

Todo Agente Gestor é criado pelo Agente Criador presente no servidor. A figura 5 ilustra o processo de criação dos Agentes especialistas de acordo com o perfil escolhido pelo usuário. Após criados os Especialistas, é iniciado o processo de montagem de carteira de ações. Apresentado no tópico seguinte.

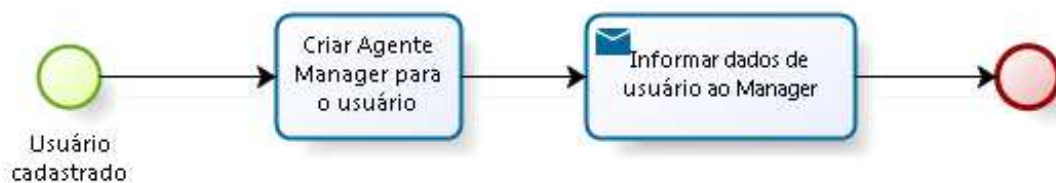


Figura 5: Processo de criação dos Agentes do tipo Gestor

• Montar carteiras de ações

A figura 6 ilustra o processo de montagem de uma carteira de ações, bem como o processo de rateio de ações entre os Agentes Especialistas vinculados ao grupo de Agentes. Vale ressaltar que de acordo com o processo previsto no capítulo 4, as ações escolhidas são alinhadas com o perfil escolhido pelo próprio usuário.

O processo de montagem de carteira de ações inicia-se no momento em que o um Agente Gestor é instanciado no servidor. Após ser instanciado, o Agente Gestor recebe do Agente criador uma mensagem contendo informações do usuário tais como: (i) Nome; (ii) Valor investido; e (iii) Perfil escolhido. O Agente Gestor envia uma mensagem ao Agente Caçador de Ações solicitando um grupo de ações compatíveis ao perfil escolhido pelo usuário, figura 7. Após escolher as ações, o Agente Gestor faz o rateio entre os Agentes Especialistas. Por fim, ao concluir os processos de montagem de carteira de ação e rateio, o Agente Gestor inicia o processo contínuo de monitoramento do risco na carteira, figura 8.

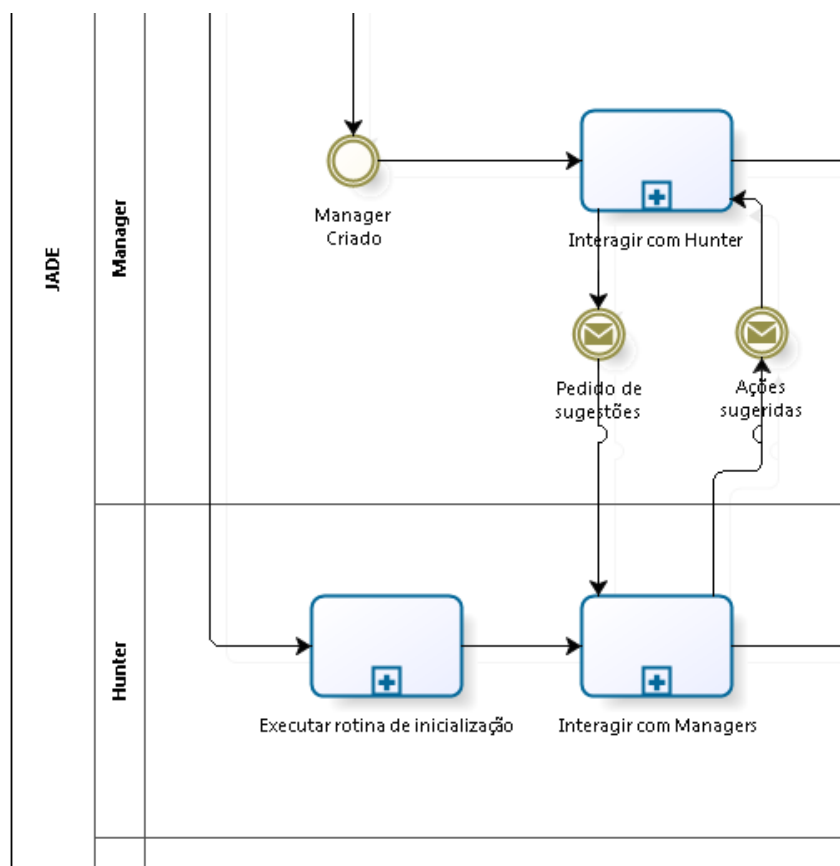
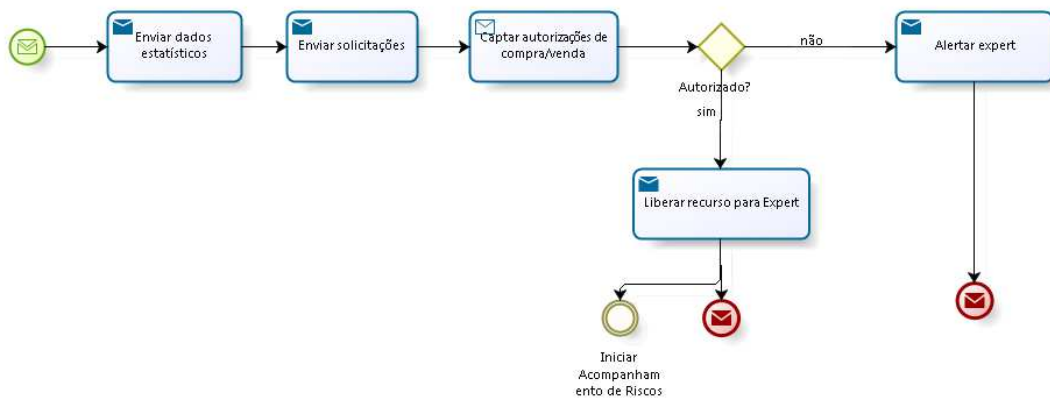


Figura 6: Processo de montagem de uma Carteira de Ações

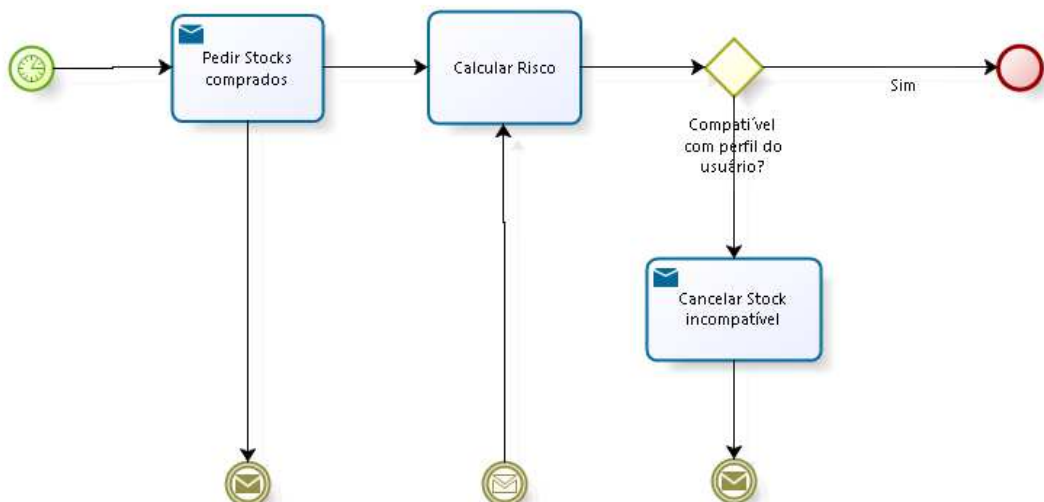
• Gerir careiras de ações

Foi implementado no Agente Gestor um comportamento cíclico responsável por



Powered by
bizagi
Modeler

Figura 7: Processo Troca de Mensagens Agente Gestor e Agente Criador



Powered by
bizagi
Modeler

Figura 8: Monitoramento de risco da carteira de ações

monitorar os riscos envolvidos na carteira de ações, que por sua vez acompanhadas por Agentes Especialistas. O Agente Gestor é capaz de cancelar uma operação de compra, caso esta represente um grau de risco incompatível com o perfil escolhido pelo usuário, figura 9.

G.3.2.1.2 Agente Especialista (Expert)

Conforme descrito no capítulo 4, o Agente Especialista é o Agente responsável por realizar leituras constantes do mercado de valores em busca de novas cotações das ações sob sua responsabilidade. Ele é criado pelo seu Agente Gestor e executa estraté-

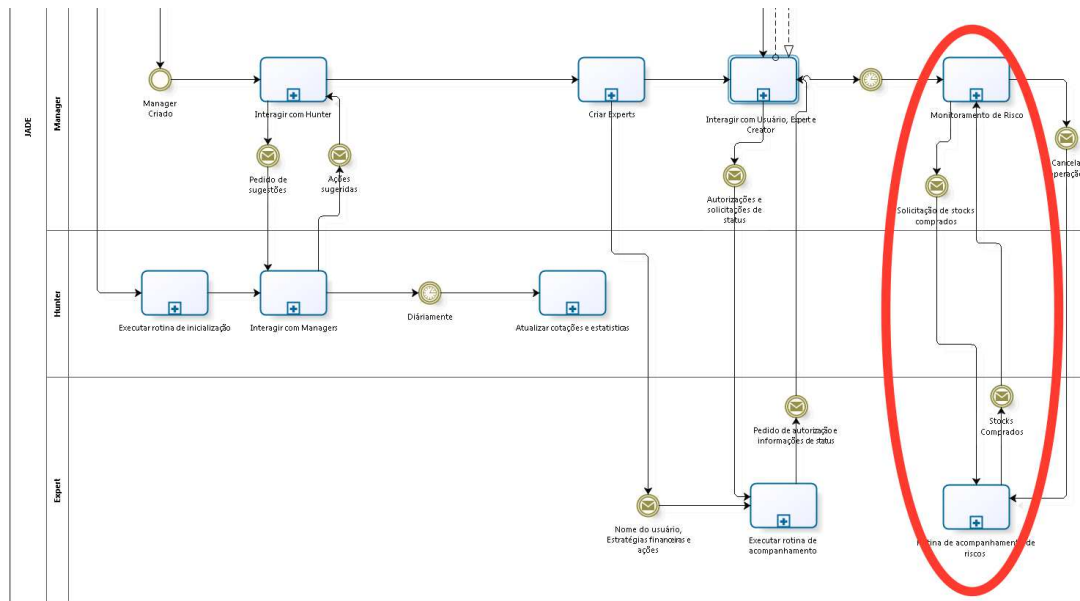


Figura 9: Acompanhamento de risco em operações

gias compatíveis com o perfil escolhido pelo usuário, recebe ainda um grupo de ações na qual será responsável. Durante o Desenvolvimento, verificou-se necessário transferir a responsabilidade de mensurar risco de estratégias para o seu Agente Gestor, e após implementação verificou-se empiricamente que executar uma rotina de simulação poderia onerar o servidor devido ao consumo de memória ao manipular objetos Candlesticks em grande quantidade. Com isso, foram implementadas as seguintes responsabilidades: (i) Aplicar estratégias que ofereçam o maior lucro possível compatível com o perfil escolhido pelo usuário; e (ii) Solicitar autorização para realizar compra ou venda de ações, figura 11.

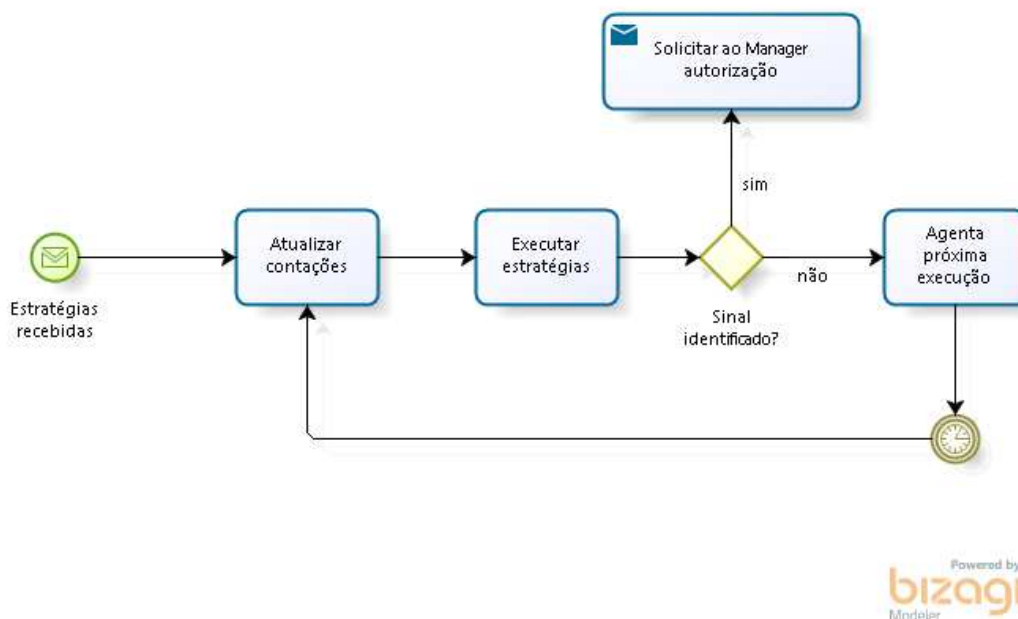


Figura 10: Acompanhamento de risco em operações

G.3.2.1.3 Agente Caçador (Hunter)

Conforme descrito no capítulo 4, o Agente Caçador é responsável por fazer buscas constantes de cotações de ações presentes na bolsa de valores e trabalhadas pela ferramenta e realizará uma categorização destas ações através de valores estatísticos, tais como: (i) retorno médio diário; (ii) retorno médio 15 e 30 dias; (iii) variância média diária; e (iv) variância média 15 e 30 dias. Através destes valores, o Agente Caçador seleciona grupos de ações compatíveis com os perfis informados por Agentes Gestores no procedimento de montagem de carteira. De acordo com as responsabilidades descritas no capítulo 4, foram implementadas as seguintes responsabilidades: (i) Busca e Seleção das melhores ações a se investir, estas ações variam de acordo com o perfil de usuário tratado pelos Agentes Gestores. O processo é ilustrado na figuras 11; e (ii) categorização de ações através de valores estatísticos, o Agente caçador faz cálculos estatísticos frequentes dada ação abordada na ferramenta e os utiliza como parâmetro de escolha para selecionar ações para atender as solicitações dos Agentes Gestores, como ilustrado na figuras 12.

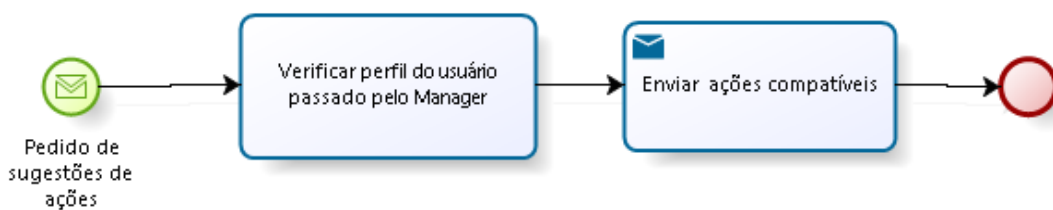


Figura 11: Processo de escolha de ações

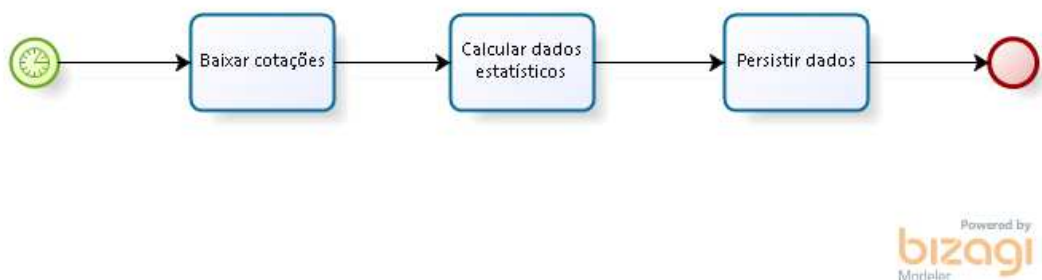


Figura 12: Persistência de valores estatísticos por ação

G.3.2.1.4 Agente Criador (Criator)

Este agente não foi previsto no capítulo 4, porém durante a implementação verificou-se a necessidade de criar um tipo de Agente para auxiliar na comunicação entre os usuários e seus respectivos grupos de Agentes. Assim como o Agente Caçador, existe apenas um agente Criador para cada servidor. Suas responsabilidades são: (i) Monitorar a criação de novas contas. O usuário quando cria uma conta o Agente Criador captura as informações repassadas pelo usuário, cria um Agente Gestor para este usuário e o informa através de

uma mensagem com informações do usuário tais como nome, valor investido e perfil escolhido; (ii) Monitorar *Log In* de usuários. Quando um usuário autentica-se na ferramenta, o Agente Criador informa ao seu Agente Gestor para que o mesmo comunique-se com seu usuário, a comunicação entre o Agente Gestor e seu respectivo usuário é demonstrado na figura 15, bem como todas funcionalidades apresentadas neste documento.

G.3.3 Pacote Suport

Este pacote comporta as classes de suporte ao funcionamento da Ferramenta de Software desenvolvida, nele estão classes relacionadas a persistência de dados e a estratégias financeiras utilizadas pelos agentes comportamentais desenvolvidos. A figura 13 ilustra de maneira geral a composição do pacote suport.

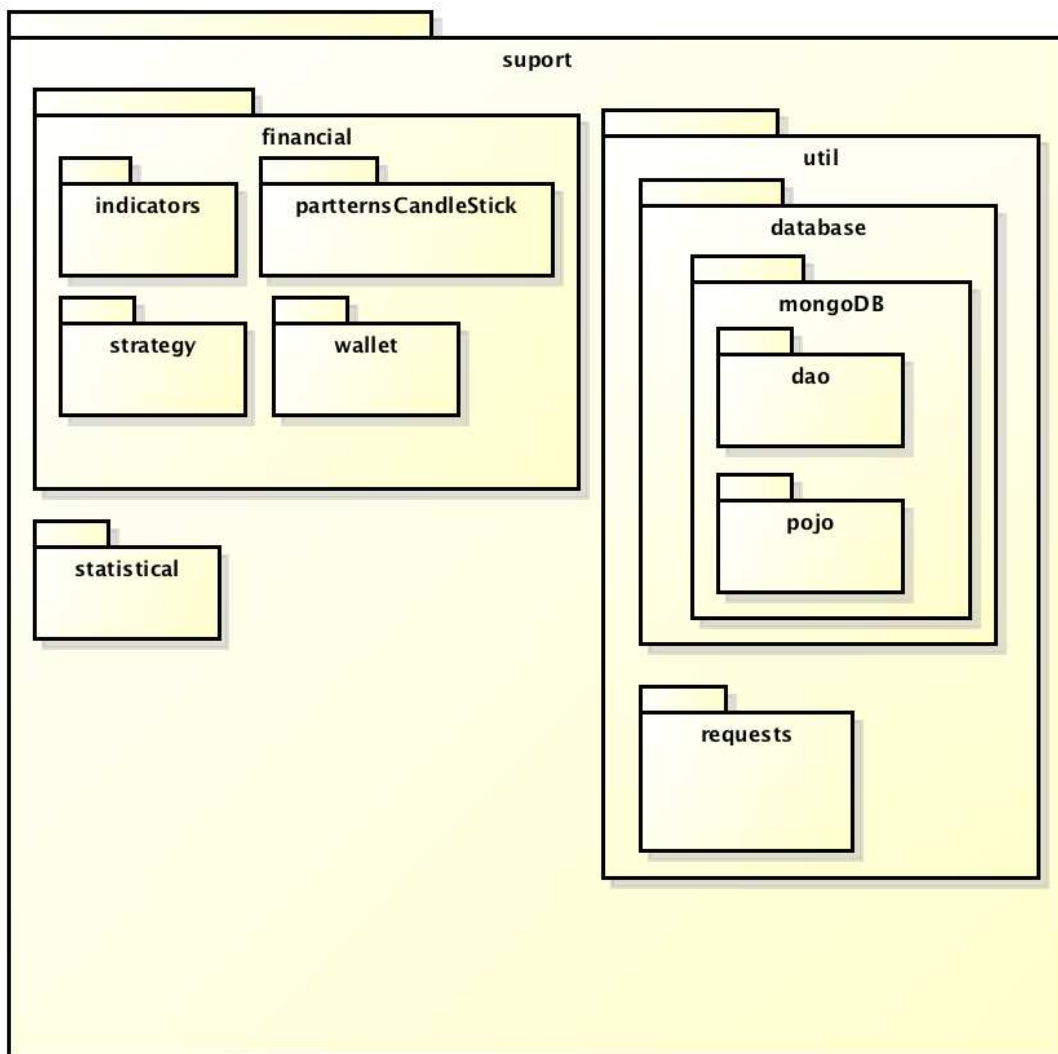


Figura 13: Pacote Suport

O pacote suport é dividido em três importantes subpacotes, financeiro, util e estatístico. No pacote financeiro estão concentradas as classes responsáveis por fornecer as estratégias financeiras utilizadas pelos agentes Especialistas. O subpacote estatístico contém classes

que dispõe de cálculos estatísticos utilizados pelo agente Caçador para encontrar Ações compatíveis com os três perfis de usuários abordados pela ferramenta. O subpacote util contém as classes responsáveis pela persistência de dados bem como requisições http de novas cotações.

O pacote suport é essencial para futuras manutenções evolutivas da ferramenta, pois para adicionar novas estratégias basta criar novas classes com estas estratégias e implementar a interface Strategy. A figura 14 apresenta o diagrama das duas principais estratégias abordadas na Ferramenta de Software. Caso a manutenção evolutiva tenha como objetivo abordar outras Bolsa de Valores ou mesmo consumir dados diretamente da BM&FBovespa, o pacote candidato a manutenção será o requests. Pois este, contém as classes que implementam a obtenção de dados e a lógica de armazenamento destes dados.

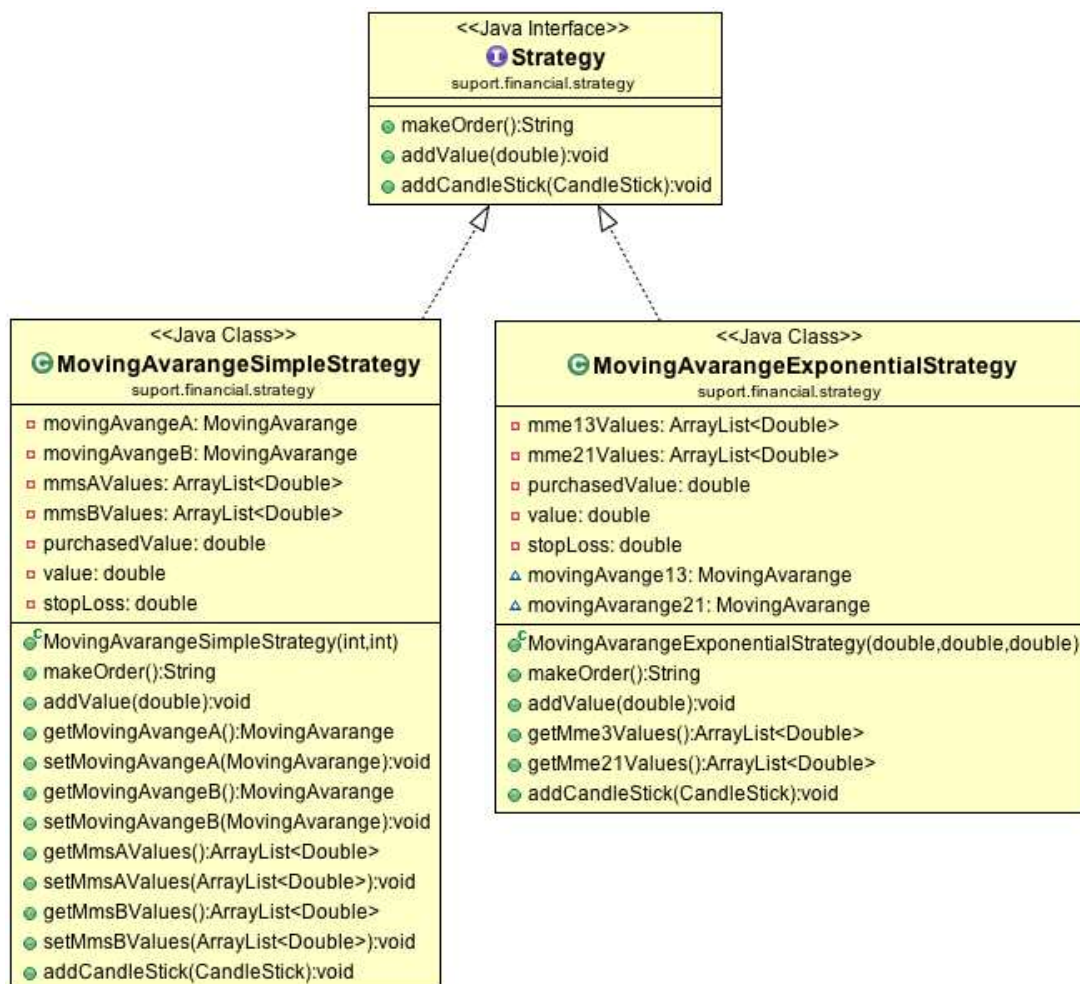


Figura 14: Estratégias Financeiras

G.4 Ferramentas

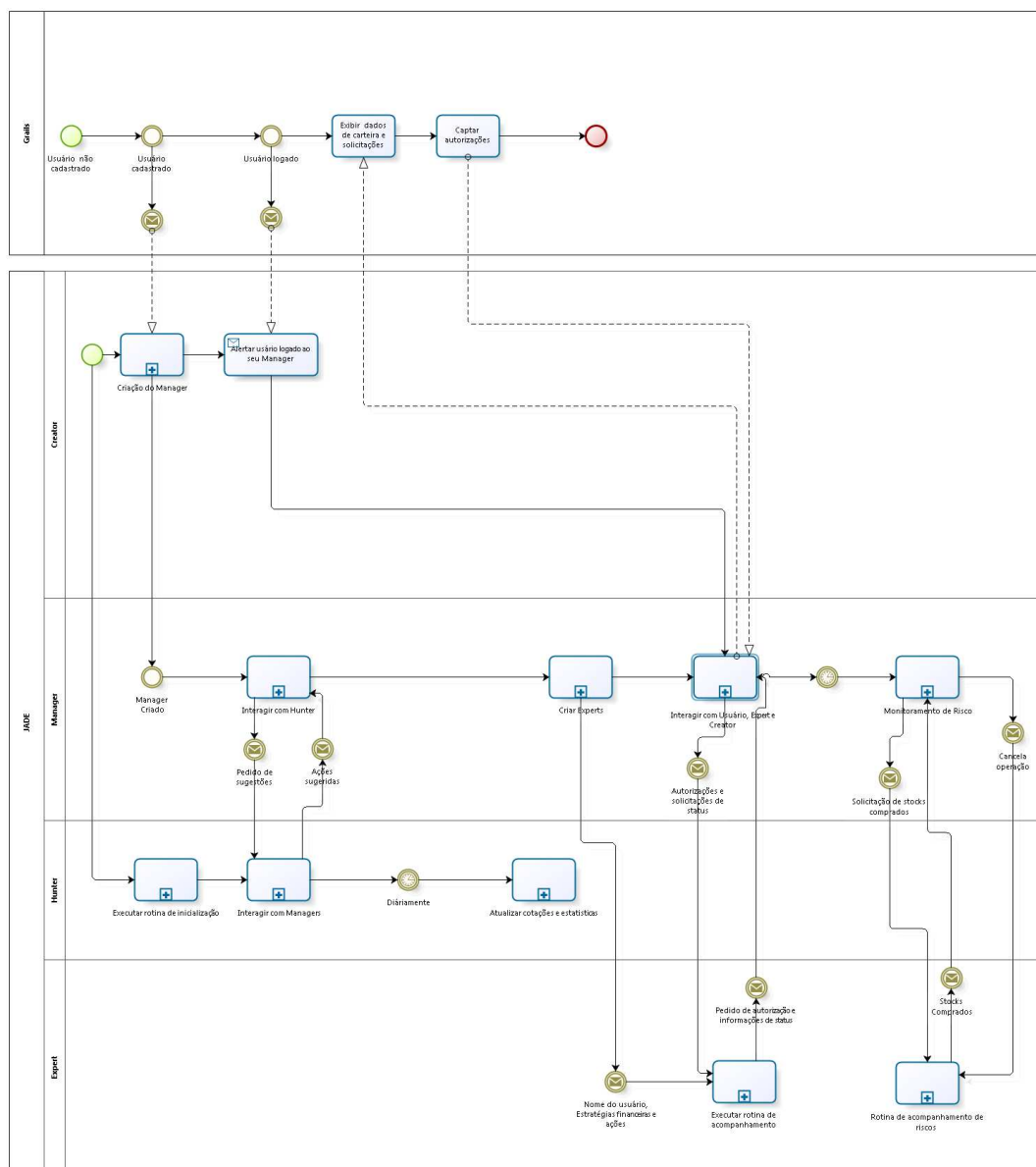


Figura 15: Visão geral da Ferramenta

Ferramentas adotadas

Nome	Versão	Responsabilidade
JADE	14.3.2	Prover Agentes Comportamentais
Grails	2.4.3	Prover Interface com usuários
MongoDB	3.0.2	Banco de Dados Não Relacional

Continuação na próxima página

Tabela 51 – *Continuação da página anterior*

Nome	Versão	Responsabilidade
Maven	3.3.3	Gerencia de dependências
Java JDK	1.7.0_79	Copilador
Eclipse Luna	4.4.0	IDE